

Titre: Analyse mathématique du problème de la frontière bilinéaire
Title: appliqué aux réseaux de neurones

Auteur: Karim Khattar
Author:

Date: 2003

Type: Mémoire ou thèse / Dissertation or Thesis

Référence: Khattar, K. (2003). Analyse mathématique du problème de la frontière bilinéaire
Citation: appliqué aux réseaux de neurones [Mémoire de maîtrise, École Polytechnique de
Montréal]. PolyPublie. <https://publications.polymtl.ca/7212/>

 **Document en libre accès dans PolyPublie**
Open Access document in PolyPublie

URL de PolyPublie: <https://publications.polymtl.ca/7212/>
PolyPublie URL:

**Directeurs de
recherche:**
Advisors:

Programme: Non spécifié
Program:

UNIVERSITÉ DE MONTRÉAL

ANALYSE MATHÉMATIQUE DU PROBLÈME DE LA FRONTIÈRE BILINÉAIRE
APPLIQUÉ AUX RÉSEAUX DE NEURONES

KARIM KHATTAR

DÉPARTEMENT DE MATHÉMATIQUES ET DE GÉNIE INDUSTRIEL
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

MÉMOIRE PRÉSENTÉ EN VUE DE L'OBTENTION
DU DIPLÔME DE MAÎTRISE ÈS SCIENCES APPLIQUÉES
(MATHÉMATIQUES APPLIQUÉES)

NOVEMBRE 2003



National Library
of Canada

Bibliothèque nationale
du Canada

Acquisitions and
Bibliographic Services

Acquisitions et
services bibliographiques

395 Wellington Street
Ottawa ON K1A 0N4
Canada

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence

ISBN: 0-612-89211-5

Our file Notre référence

ISBN: 0-612-89211-5

The author has granted a non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.

The author retains ownership of the copyright in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

L'auteur conserve la propriété du droit d'auteur qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

In compliance with the Canadian Privacy Act some supporting forms may have been removed from this dissertation.

Conformément à la loi canadienne sur la protection de la vie privée, quelques formulaires secondaires ont été enlevés de ce manuscrit.

While these forms may be included in the document page count, their removal does not represent any loss of content from the dissertation.

Bien que ces formulaires aient inclus dans la pagination, il n'y aura aucun contenu manquant.

Canada

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Ce mémoire intitulé :

ANALYSE MATHÉMATIQUE DU PROBLÈME DE LA FRONTIÈRE BILINÉAIRE
APPLIQUÉ AUX RÉSEAUX DE NEURONES

présenté par : KHATTAR Karim

en vue de l'obtention du diplôme de : Maîtrise ès sciences appliquées

a été dûment accepté par le jury d'examen constitué de :

M. BRAULT Jean-Jules, Ph.D., président

M. LABIB Richard, Ph.D., membre et directeur de recherche

M. ADJENGUE Luc D., Ph.D., membre et codirecteur de recherche

M. ROUSSELLE Jean, Ph.D., membre

À mes parents, à ma soeur

REMERCIEMENTS

J'aimerais exprimer ma gratitude envers mon directeur de recherche, le docteur Richard Labib, qui a toujours été disponible pour répondre à toutes les questions lui étant soumises. Sa rigueur intellectuelle, ses conseils pertinents et son appui continu m'ont permis de réaliser ce travail.

J'aimerais également remercier Nada Ebeid sans qui l'aboutissement de ce mémoire n'aurait pas pu avoir lieu. Je voudrais aussi la remercier pour son continuel appui et ses encouragements.

Je suis reconnaissant envers mon cousin Antoine Chémali pour ces judicieux conseils.

Enfin je voudrais remercier ma famille (ma cousine Reem) et tous mes amis qui m'ont soutenu de proche ou de loin durant ce travail.

RÉSUMÉ

Ce mémoire est une contribution au domaine des réseaux de neurones appliqué aux problèmes de classification et d'approximation de fonctions. La fonction que nous voulons étudier est une frontière bilinéaire, c'est-à-dire qu'elle est formée par deux demi-droites sécantes.

Le réseau utilisé appartient à la famille des MLP (Multilayer Perceptrons). Il est formé d'une première couche à deux entrées, une couche cachée à deux neurones et une couche de sortie ayant un seul neurone. Les sorties des neurones d'entrées définissent un nouvel espace que nous appelons l'espace caché.

Dans un premier temps, l'étude est orientée vers la séparation de deux classes de données. À l'aide d'une analyse dans l'espace caché, nous obtenons des résultats permettant de savoir sous quelles conditions le réseau est capable de parfaitement séparer deux classes séparables par une frontière bilinéaire. L'importance de la valeur du paramètre de la fonction d'activation des neurones est mise en évidence.

Dans un deuxième temps, et dans le but d'approfondir l'étude du réseau, une fonction explicite, représentant la frontière générée par le réseau, est obtenue. L'analyse de cette nouvelle fonction permet l'obtention de diverses informations relatives à l'effet des paramètres du réseau sur la frontière qu'il génère. Nous verrons aussi qu'il est possible

de généraliser ce résultat et d'obtenir une équation explicite même en augmentant, jusqu'à quatre, le nombre de neurones cachés dans ce réseau.

Finalement, et dans le but d'optimiser le réseau afin qu'il reproduise le mieux possible une frontière bilinéaire, nous définissons une expression représentant l'erreur d'approximation. Nous cherchons à minimiser cette erreur et trouver les valeurs optimales des paramètres.

ABSTRACT

This paper is a contribution to the field of neural networks. It provides an innovative solution to the problem of classification and function approximation. The function we want to approximate is a bilinear boundary, since it is formed by two semi-lines intersecting at one point.

The network developed in this study is a multilayer perceptron (MLP) formed by the combination of two inputs, two hidden neurons, and one output. The outputs of the hidden neurons form what we call the hidden space.

First, we will study the network's capacity to isolate two classes of data that can be separated by a bilinear frontier. The analysis of the hidden space will provide results on how and when the network is capable of forming a perfect separation of these two classes. The activation function used by the network is the sigmoid, and the importance of the value of its parameter is highlighted.

Second, we will obtain an explicit analytical function describing the boundary generated by the network. A thorough analysis of this function will provide information on the effect each parameter has on the network's behavior. We will also see how to generalize the results in order to get an explicit function even with additional neurons in the hidden layer of the network.

Finally, we want to optimize the network in order to make it reproduce the best possible bilinear boundary. The ultimate purpose is to optimize the parameters of the network in order to achieve minimum error.

TABLE DES MATIÈRES

DÉDICACE.....	iv
REMERCIEMENTS.....	v
RÉSUMÉ.....	vi
ABSTRACT	viii
TABLE DES MATIÈRES	x
LISTE DES FIGURES.....	xiii
INTRODUCTION	1
CHAPITRE 1 SÉPARATION DE DEUX CLASSES.....	6
1.1 Frontière doublement linéaire.....	6
1.2 Réseau à un seul neurone	8
1.3 Réseau à deux couches et deux neurones.....	10
1.4 Fonction d'activation	13
1.4.1 Sigmoides.....	13
1.4.2 Transformation de l'espace des entrées.....	15
1.4.3 Fonction Heaviside.....	17
1.5 Réseau à trois couches et trois neurones.....	19
1.5.1 Lemme 1	20
1.6 Étude pratique de la fonction sigmoïde.....	21

1.6.1 Proposition 1	23
1.7 Théorème 1	30
CHAPITRE 2 ÉQUATION EXPLICITE.....	37
2.1 Équation explicite de la frontière.....	37
2.1.1 Lemme 2	37
2.1.2 Intersection entre les droites et la surface de séparation	46
2.1.3 Frontière du réseau et asymptotes.....	57
2.2 Généralisation du résultat.....	60
2.2.1 Lemme 3	61
2.2.2 Théorème 3	65
2.3 Justification du choix du repère.....	66
2.3.1 Changement de repère	69
CHAPITRE 3 OPTIMISATION DES PARAMÈTRES	78
3.1 Pourcentage d'erreur dans l'espace caché.....	79
3.1.1 Étude de la fonction de densité dans l'espace caché.....	80
3.1.2 Étude de l'espace image	83
3.2 Pourcentage d'erreur dans le plan des entrées.....	84
3.3 Fonction explicite symétrique	87
3.3.1 Optimisation de la surface d'erreur.....	89
3.4 Fonction explicite simplifiée	96
3.4.1 Intégrale de la fonction simplifiée	100
3.5 Méthodes d'optimisation.....	101

CONCLUSION.....	103
BIBLIOGRAPHIE.....	105

LISTE DES FIGURES

Figure 1.1	Frontière doublement linéaire	6
Figure 1.2	Courbe de variation de la conductance en fonction du volume de soude..	7
Figure 1.3	Réseau formé d'un neurone	8
Figure 1.4	Séparation linéaire avec un seul neurone.....	9
Figure 1.5	Réseau à deux neurones.....	11
Figure 1.6	Séparation d'un plan par deux droites	12
Figure 1.7	Fonction Sigmoidale	14
Figure 1.8	Espace caché	16
Figure 1.9	Fonction Heaviside.....	18
Figure 1.10	Espace caché pour a égale l'infini.....	18
Figure 1.11	Réseau à deux entrées, une sortie et deux neurones cachés.....	19
Figure 1.12	Différentes possibilités pour la droite d_3	20
Figure 1.13	Allure de la fonction sigmoïde pratique	23
Figure 1.14	Données et surface de séparation	25
Figure 1.15	Séparation parfaite de données.....	28
Figure 1.16	Deux demi-plans positifs et deux autres négatifs.....	30
Figure 1.17	Espace caché et recherche du a minimum	31
Figure 1.18	Différentes données dans l'espace des entrées	34
Figure 1.19	Espace caché pour $a = 1$	34
Figure 1.20	Espace caché pour $a = 3.3$	35

Figure 2.1	Courbe générée par le réseau	45
Figure 2.2	Droite d_3 sans intersection avec d'_1 et d'_2	49
Figure 2.3	Exemple de frontière sans intersection.....	49
Figure 2.4	Droite d_3 avec deux intersections.....	50
Figure 2.5	Exemple de frontière avec deux intersections.....	51
Figure 2.6	Frontière sans points d'intersection et $a = 1$	54
Figure 2.7	Frontière sans points d'intersection et $a = 5$	54
Figure 2.8	Frontière avec un seul point d'intersection.....	55
Figure 2.9	Séparation des données avec le a minimum	56
Figure 2.10	Asymptotes et espace caché.....	57
Figure 2.11	Droites de pentes de même signe	60
Figure 2.12	Réseau à trois neurones cachés	61
Figure 2.13	Frontière qui n'est pas une fonction: droites de pentes opposés.....	67
Figure 2.14	Frontière qui n'est pas une fonction : droites de pentes de même signe .	67
Figure 2.15	Droite à pente positive et signe des demi-plans.....	70
Figure 2.16	Droite à pente négative et signe des demi-plans	70
Figure 2.17	Union de deux demi-plans positifs.....	71
Figure 2.18	Translation d'un repère.....	73
Figure 2.19	Rotation avec deux droites à pentes positives.....	74
Figure 2.20	Rotation avec deux droites à pentes négatives.....	75
Figure 2.21	Rotation de $-\pi/2$	75
Figure 2.22	Rotation de $\pi/2$	76

Figure 2.23	Séparation des deux demi-plans négatifs.....	76
Figure 3.1	Surface d'erreur et espace de définition	78
Figure 3.2	Espace caché et surface d'erreur	79
Figure 3.3	Espace de variation R_1	80
Figure 3.4	Espace image avec $a = 1$	81
Figure 3.5	Espace image avec $a = 3$	81
Figure 3.6	Espace $x \in [6.4, 8.4]$ et $y \in [0.25, 2.25]$	83
Figure 3.7	Délimitation de l'erreur (E) dans l'espace image pour $a = 1$	84
Figure 3.8	Bornes définissant la surface d'erreur	85
Figure 3.9	Droites symétriques par rapport aux ordonnées.....	88
Figure 3.10	Surface d'erreur pour frontière quelconque.....	89
Figure 3.11	Frontière bilinéaire symétrique et domaine de variation	92
Figure 3.12	Courbe d'erreur.	93
Figure 3.13	Valeurs de d'_1 , d'_2 et d'_3 en fonction de b_3	94
Figure 3.14	Courbe optimale pour l'exemple 3.1	95
Figure 3.15	Espace des entrées avec d_1 confondu avec l'axe des ordonnées.....	97
Figure 3.16	Courbe obtenue avec l'équation simplifiée 3.9.....	100

INTRODUCTION

L'homme s'est souvent inspiré de la nature afin de pouvoir mener à bien ses propres créations. Alors quoi de mieux que le cerveau humain en guise d'inspiration pour celui qui veut créer de l'intelligence artificielle? En effet, le cerveau humain est une entité biologique fort complexe dont la puissance et l'étendue ne sont pas à mettre en doute. C'est donc dans le but ultime de pouvoir un jour reproduire artificiellement le cerveau que les réseaux de neurones ont vu le jour en 1943. À partir de ce moment, et pendant plusieurs années, ce domaine fut l'objet de nombreuses recherches. Cependant en 1969, Minsky et Papert publiaient une monographie qui semblait prouver l'inutilité des réseaux de neurones. Cette publication fut un des facteurs qui a amené beaucoup de chercheurs à se désintéresser de ce domaine. À partir des années 80, après diverses publications par plusieurs auteurs, l'intérêt pour les réseaux de neurones réapparut et est resté en continuelle croissance.

Depuis, les réseaux de neurones sont utilisés dans différents domaines, et pour plusieurs types de problèmes. Dans ce travail nous faisons une analyse mathématique détaillée du problème de classification et d'approximation de frontières bilinéaires. Les problèmes de classifications sont connus, et se retrouvent dans plusieurs applications. Par exemple, lorsque nous recensons des données et que nous voulons les séparer suivant certains critères alors nous traitons un problème de classification. Il en va de même pour les problèmes d'approximation de frontières, que nous retrouvons par exemple pour faire de

la modélisation. Dans plusieurs domaines nous pouvons trouver des applications au type de frontière que nous étudions. En chimie, par exemple, les courbes représentant l'évolution de la conductance d'une solution acide, à laquelle nous ajoutons une solution basique, ont la forme d'une frontière bilinéaire; et encore, les courbes représentant le titrage d'un acide par une base ou d'une base par un acide peuvent être approximées par l'union de deux frontières bilinéaires.

Les réseaux de neurones étant une tentative de reproduire artificiellement un cerveau humain, il existe en conséquence plusieurs analogies entre le mode de fonctionnement de ces deux entités. En effet, afin de tenter de reproduire la propagation des signaux électrochimiques dans le cerveau, des vecteurs de données sont traités dans un réseau de neurones artificiels. Chaque neurone peut avoir un signal ou plusieurs signaux d'entrées et un signal ou plusieurs signaux de sortie. Dans la littérature, il est possible de retrouver les différentes équations qui ont été définies et qui permettent de trouver les liens entre ces entrées et ces sorties. Ces équations seront le point de départ de notre étude qui utilisera un réseau de type MLP (MultiLayer Perceptrons). Il existe cependant d'autres types de réseaux tels les RBF (Radial-Basis Function Networks), les SVM (Support Vector Machine), etc.

Il a déjà été démontré qu'un MLP est capable de reproduire n'importe quelle séparation non linéaire, à condition d'avoir un nombre suffisant de neurones. Il est aussi connu qu'un réseau à une seule couche cachée est capable de reproduire n'importe quelle

fonction avec le degré de précision désiré, à condition d'ajouter le nombre de neurones nécessaires dans la couche cachée. Diverses études se sont penchées sur le choix de la fonction d'activation d'un réseau. Une fonction d'activation est une fonction quelconque, généralement non linéaire, qui définit la sortie d'un neurone en fonction de ses entrées (Haykin, 1999). Il a été montré qu'avec un choix adéquat de la fonction d'activation, un réseau à une seule couche est capable de résoudre certains problèmes non linéaires dont celui du ou-exclusif (Zhengwen et Sarhadi, 1993).

Parmi les différentes recherches qui se rapportent aux problèmes des frontières et des réseaux neuronaux, il n'y en a que très peu qui traitent directement le problème de frontière doublement linéaire; notons l'article de Chulhee et Eunsuk (2000) et celui de Chulhee et Kwon (2001). Dans le premier de ces documents les auteurs présentent, pour des réseaux à trois couches, l'effet des neurones cachés, l'aspect de non linéarité qu'ils introduisent et leurs effets sur la dimensionnalité de l'espace caché. Différents types de frontières, dont une bilinéaire, sont aussi illustrées. Dans le second document, les auteurs s'attardent sur l'apprentissage de la représentation de la frontière d'un objet. Pour ce faire, ils génèrent des données de part et d'autre de cette frontière et utilisent ces données pour l'apprentissage du réseau. L'une des frontières qu'ils proposent est doublement linéaire. Elle est obtenue à l'aide d'un réseau à deux neurones d'entrées, trois neurones cachés et deux neurones de sorties.

Cependant, à la différence des deux articles que nous venons de citer, notre étude est

une étude théorique, analytique et détaillée des frontières bilinéaires et d'un problème de classification qui lui est relié. En effet, dans un premier temps, nous montrerons comment un réseau, très simple, à deux neurones d'entrées, deux neurones cachés et un seul de sortie, est capable de séparer deux classes séparables par une frontière bilinéaire. Nous mettrons aussi en évidence l'importance de la valeur du paramètre de la fonction d'activation. Dans un deuxième temps, nous développerons une fonction nouvelle de la forme $y = f(x)$. Cette fonction explicite permettra de tracer la frontière générée par le réseau, elle sera indispensable pour étudier avec exactitude le comportement de cette frontière. Nous verrons aussi qu'il est possible de généraliser le résultat obtenu et de trouver une fonction explicite même en ajoutant un certain nombre de neurones dans la couche cachée. Ensuite, à l'aide de la fonction explicite nous trouverons une expression générale de l'erreur que nous cherchons à minimiser. Finalement, grâce à de remarquables simplifications nous serons en mesure de simplifier la fonction explicite trouvée.

Le but final de cette étude exhaustive est de trouver les paramètres optimaux du réseau. C'est-à-dire les paramètres qui vont permettre de minimiser l'erreur entre la frontière générée par le réseau et la fonction bilinéaire à approximer. Une fois cette valeur minimale obtenue, elle peut nous être très utile puisqu'elle sert de référence. Nous pouvons l'utiliser pour évaluer les performances d'un algorithme d'apprentissage. Si par exemple, l'algorithme d'apprentissage appelé algorithme de rétro propagation permet d'obtenir une performance comparable à la performance optimale théorique, il

n'est peut être pas nécessaire de chercher un autre algorithme qui serait plus complexe et qui impliquerait une augmentation du temps d'apprentissage. Si par contre la performance obtenue est considérablement inférieure à celle de la référence, il est probablement utile de chercher une solution alternative. Nous pouvons aussi utiliser ce pourcentage optimal afin de comparer ce réseau à un réseau d'une autre famille (exemple RBF), mais qui a le même nombre de paramètres. L'étude théorique que nous faisons va donc permettre d'obtenir des résultats qui serviront à évaluer des performances pratiques.

Tout au long de ce mémoire, et dans le but de clarifier les différents lemmes et théorèmes obtenus, nous présenterons des exemples clairs et intéressants.

CHAPITRE 1 SÉPARATION DE DEUX CLASSES

1.1 Frontière doublement linéaire

Nous allons commencer par expliquer ce qu'est une frontière doublement linéaire.

Soit un plan xy et deux classes différentes de données. L'une de ces classes forme l'ensemble des données que nous nommons C_1 , tandis que l'autre classe forme l'ensemble nommé C_2 . Supposons que les données provenant de la classe C_1 soient représentées par des cercles (\bigcirc) et ceux de la classe C_2 soient représentées par des croix (\times); alors ces cercles et ces croix peuvent être disposés d'une infinité de manières dans le plan xy . Cependant, dans le cadre de cette étude, nous nous intéressons à un cas particulier: le cas où C_1 et C_2 peuvent être séparés par une frontière doublement linéaire.

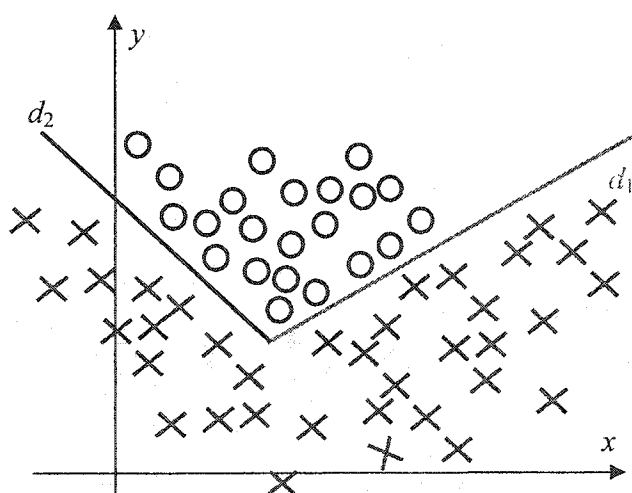


Figure 1.1 Frontière doublement linéaire

Nous voyons, dans la figure 1.1, que les deux classes C_1 et C_2 peuvent être séparées à l'aide des deux demi-droites d_1 et d_2 qui se coupent en un point. La séparation formée par ces deux droites est ce que nous appelons une frontière doublement linéaire ou bilinéaire.

Comme nous l'avons déjà mentionné, nous pouvons trouver plusieurs exemples de frontières bilinéaire dans le domaine de la chimie, en voici un.

Soit une solution d'acide chlorhydrique à laquelle nous avons ajoutée de l'eau.

Si nous notons la valeur de la conductance après avoir procédé à des additions répétées de quantités connues d'ions hydroxyde. Voici la courbe que nous obtenons

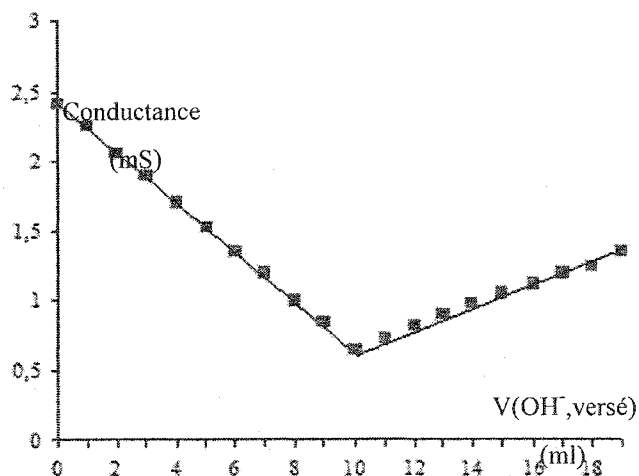


Figure 1.2 Fonction de variation de la conductance en fonction du volume de soude

La frontière que nous voyons dans la figure ci-dessus peut être approximée par une fonction bilinéaire comme celle de la figure 1.1.

1.2 Réseau à un seul neurone

La base de tout réseau est un simple neurone. La représentation mathématique et les différentes équations qui gouvernent un neurone sont citées ci-dessous.

La figure 1.3 représente un seul neurone avec :

- Deux entrées x et y tel que $(x, y) \in \mathfrak{R}$
- Une sortie S
- Deux poids w_1 et w_2
- Un biais b
- Une fonction d'activation $\varphi(.)$
- Une sortie linéaire V_s

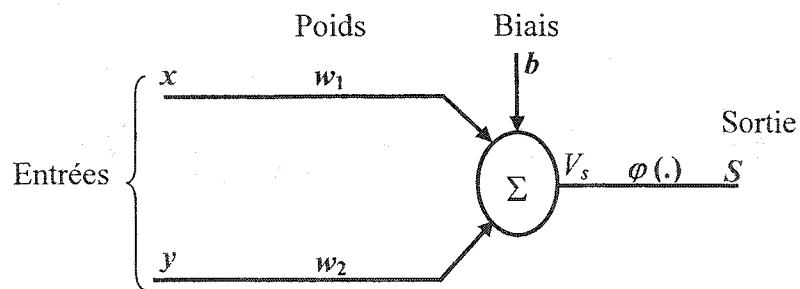


Figure 1.3 Réseau formé d'un neurone

Les équations mathématiques pour ce neurone sont:

$$V_s = w_1x + w_2y + b$$

$$S = \varphi(V_s)$$

Soit un plan xy avec des données provenant des deux classes C_1 et C_2 comme le montre la figure 1.4.

Soit aussi une droite d_1 d'équation :

$$y = -W_1x - B \Rightarrow W_1x + y + B = 0$$

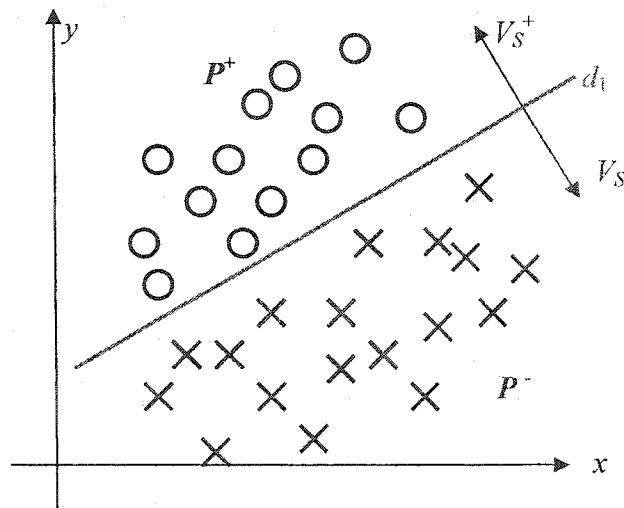


Figure 1.4 Séparation linéaire avec un seul neurone

En posant $w_1 = W_1$, $w_2 = 1$, et $b = B$, nous obtenons l'équation de la sortie linéaire :

$$V_s = w_1x + y + b$$

Cette dernière équation est identique à celle de d_1 . La droite d_1 coupe le plan en deux demi-plans : P^+ et P^- . Dans le demi-plan P^+ la sortie V_s est positive, dans le demi-plan P^- la sortie V_s est négative et le long de la droite, V_s est nulle. Un seul neurone est donc capable de séparer des données linéairement séparables, le signe de la sortie linéaire V_s

du réseau permet d'identifier le demi-plan auquel le vecteur d'entrée appartient.

Note:

- Le rôle du biais est de générer des frontières qui ne passent pas nécessairement par l'origine.
- Dans notre cas nous faisons une étude en 2 dimensions x et y . Nous étudions donc le cas d'une droite dans un plan. Cependant, dans le cas où nous travaillons avec plus que deux dimensions, il est possible, avec un neurone ayant plus que deux entrées, de tracer n'importe quel hyperplan.

1.3 Réseau à deux couches et deux neurones

Sachant qu'un perceptron formé d'un neurone est capable de tracer n'importe quelle droite, il est normal d'avoir recours à un réseau à deux neurones afin de former deux droites dans un plan.

Soit deux demi-droites d_1 et d_2 , formant une séparation bilinéaire, d'équations respectives

$$\begin{cases} y = -W_{11}x - B_1 \Rightarrow W_{11}x + y + B_1 = 0 \\ y = -W_{12}x - B_2 \Rightarrow W_{12}x + y + B_2 = 0 \end{cases} \quad 1.1$$

Soit le réseau suivant :

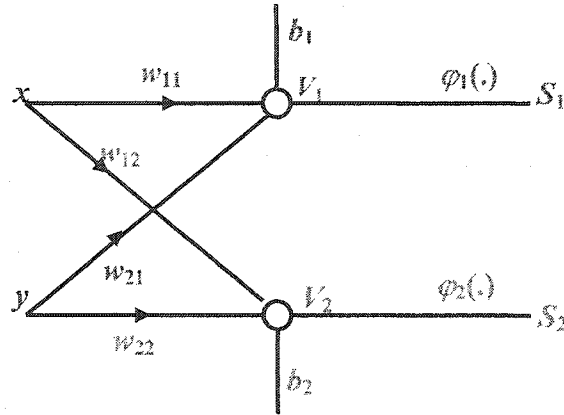


Figure 1.5 Réseau à deux neurones

Le réseau à deux neurones de la figure 1.5 ci-dessus devrait nous permettre de tracer les deux droites d_1 et d_2 . En effet, V_1 et V_2 étant respectivement les sorties linéaires du premier et du deuxième neurone, nous pouvons écrire les équations suivantes :

$$\begin{cases} V_1 = w_{11}x + w_{12}y + b_1 \\ V_2 = w_{21}x + w_{22}y + b_2 \end{cases} \quad 1.2$$

$$\begin{cases} S_1 = \phi_1(V_1) \\ S_2 = \phi_2(V_2) \end{cases} \quad 1.3$$

Avec l'équation 1.1 et en fixant

$$W_{11} = w_{11}, W_{12} = 1 \text{ et } B_1 = b_1$$

$$W_{21} = w_{21}, W_{22} = 1 \text{ et } B_2 = b_2$$

Nous retrouvons alors les équations des deux droites que voici :

$$\begin{cases} V_1 = w_{11}x + y + b_1 \\ V_2 = w_{21}x + y + b_2 \end{cases} \quad 1.4$$

Les deux grandeurs V_1 et V_2 permettent de définir deux demi-plans positifs P_1^+ et P_2^+ , et deux autres négatifs P_1^- et P_2^- (Figure 1.6) de sorte que :

- $V_1 > 0$ dans P_1^+
- $V_1 < 0$ dans P_1^-
- $V_2 > 0$ dans P_2^+
- $V_2 < 0$ dans P_2^-

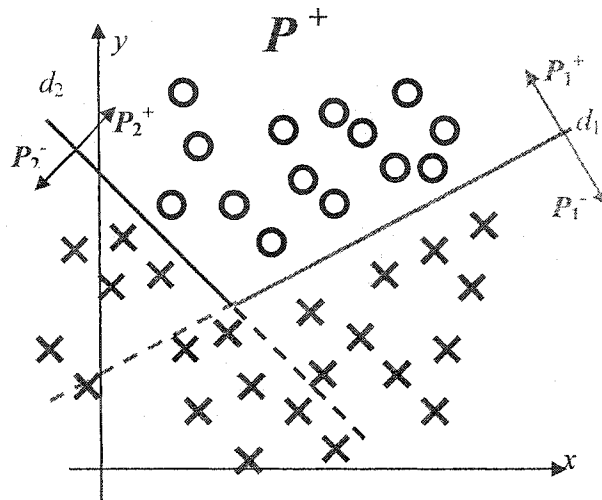


Figure 1.6 Séparation d'un plan par deux droites

Note :

- Nous appelons P^+ l'intersection des 2 demi-plans P_1^+ avec P_2^+ .
- Nous supposons toujours que la classe C_1 appartient à P^+ et C_2 appartient au reste du plan.

Nous étudions le cas où nous voulons reproduire une frontière bilinéaire qui est une

fonction, c'est-à-dire une valeur de y pour une valeur de x . Nous verrons plus loin comment nous pouvons toujours nous ramener à une frontière respectant cette contrainte.

Étudions maintenant les sorties S_1 et S_2 , pour ce faire nous allons analyser la fonction d'activation φ .

1.4 Fonction d'activation

Des résultats mathématiques ont déjà montré que pour qu'un réseau de neurones génère des frontières plus complexes que des frontières linéaires, il faut utiliser des fonctions d'activation non linéaires. Parmi les différentes fonctions non linéaires auxquelles nous pouvons avoir recours, la sigmoïde a des propriétés très intéressantes. C'est d'ailleurs avec cette fonction très communément utilisée dans les réseaux de neurones que nous allons travailler.

1.4.1 Sigmoïde

Voici l'équation et la représentation d'une fonction Sigmoïde :

$$\varphi(x) = \frac{1}{1 + e^{-ax}} \quad 1.5$$

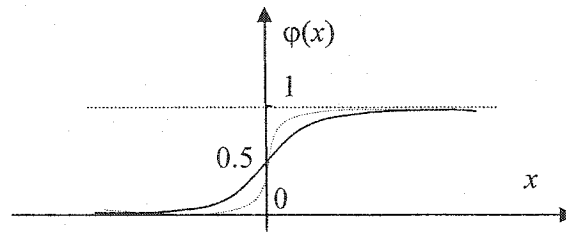


Figure 1.7 Fonction Sigmoide

Note:

- a est appelé le paramètre de la sigmoïde, il est toujours positif. La variation de ce paramètre permet de modifier la courbure de la fonction (Figure 1.7).
- Nous considérons que les fonctions d'activation des différents neurones d'un même réseau sont égales. Donc dans notre cas que $\varphi_1 = \varphi_2 = \varphi$.

Caractéristiques de la sigmoïde

$$\varphi(x) \in]0, 1[$$

$$\varphi(x) > 0.5 \text{ si } x > 0$$

$$\varphi(x) < 0.5 \text{ si } x < 0$$

$$\varphi(x) = 0.5 \text{ si } x = 0$$

Une sigmoïde a deux asymptotes, en effet :

$$\lim_{x \rightarrow -\infty} \varphi(x) = 0$$

$$\lim_{x \rightarrow +\infty} \varphi(x) = 1$$

1.4.2 Transformation de l'espace des entrées

Sachant que $\varphi_1 = \varphi_2$ et en combinant les équations 1.3 et 1.5 nous obtenons :

$$\left\{ \begin{array}{l} S_1 = \frac{1}{1 + e^{-aV_1}} \\ S_2 = \frac{1}{1 + e^{-aV_2}} \end{array} \right. \quad 1.6$$

Nous pouvons donc dire que S_1 est l'image de V_1 par la fonction $\varphi(\cdot)$ et S_2 est l'image de V_2 par cette même fonction. Le plan S_1S_2 est donc l'image du plan xy . Ce nouveau plan S_1S_2 que nous nommons *espace caché* va s'avérer très utile pour l'étude du réseau. En effet, plusieurs de nos résultats vont découler d'analyses faites dans cet espace, il est donc important que nous l'analysions en détail.

Sachant quelles sont les caractéristiques d'une sigmoïde nous pouvons conclure que

$$(S_1, S_2) \in]0, 1[$$

L'image de chacun des points du plan xy se situe dans un carré unitaire tel que nous le voyons ci-dessous. Tout le plan xy est réduit à ce carré unitaire (excluant les côtés).

Nous appelons les points dans l'espace caché les *points images*.

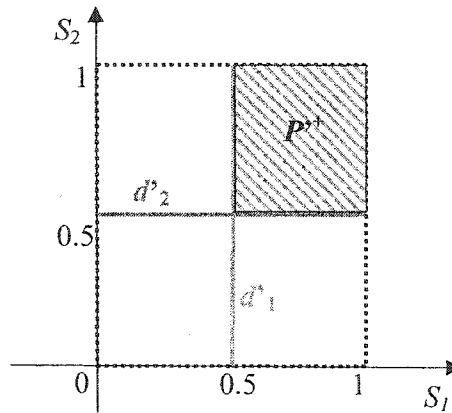


Figure 1.8 Espace caché

Nous voulons trouver les correspondances entre l'espace caché (Figure 1.8) et l'espace des entrées (Figure 1.6).

Le long de la droite d_1 (Figure 1.6):

$$V_1 = 0 \text{ et } V_2 \text{ varie de } -\infty \text{ à } +\infty$$

$$\{V_1 = 0, V_2 \in \mathbb{R}\} \Rightarrow \{S_1 = \varphi(V_1) = 0.5, S_2 = \varphi(V_2) \in]0, 1[\}.$$

Le segment de droite d'_1 dans l'espace caché est défini par :

$$\{S_1 = 0.5 \text{ et } S_2 \in]0, 1[\}.$$

Nous pouvons donc conclure que l'image de la droite d_1 du plan xy est le segment d'_1 dans le plan S_1S_2 . Avec une analyse similaire nous pouvons montrer que le segment d'_2 est l'image de la droite d_2 .

Analysons les demi-plans générés par les deux droites (d_1 et d_2).

Nous avons un demi-plan P_1^+ où :

- $V_1 \geq 0 \quad \Rightarrow \quad S_1 \geq 0.5$
- $-\infty < V_2 < +\infty \quad \Rightarrow \quad S_2 \in]0, 1[$

Et un demi-plan P_2^+ où :

- $V_2 \geq 0 \quad \Rightarrow \quad S_2 \geq 0.5$
- $-\infty < V_1 < +\infty \quad \Rightarrow \quad S_1 \in]0, 1[$

Ainsi, l'image de P^+ défini par $(V_1 \text{ et } V_2) \geq 0$, est le carré hachuré P'^+ , à l'intérieur duquel $(S_1 \text{ et } S_2) \geq 0.5$ (Figure 1.8). Le carré hachuré de la figure 1.8 est donc la transformation de la partie du plan P^+ de la figure 1.6. Pour reproduire la frontière bilinéaire nous devons isoler P^+ du reste du plan, ce qui correspond à isoler les points images dans le carré hachuré, P'^+ , du reste des points images. De même, pour isoler les données contenues dans P^+ du reste des données, il suffit d'isoler leurs points images contenus dans P'^+ , du reste des points images.

1.4.3 Fonction Heaviside

Nous avons vu que le paramètre a de la sigmoïde affecte la courbure de cette dernière. Si ce paramètre devient égal à l'infini, nous avons alors un cas particulier, la fonction sigmoïde devient la fonction Heaviside donnée par la figure 1.9 ci-dessous.

Nous avons :

$$\varphi(x) = 1 \text{ si } x \geq 0$$

$$\varphi(x) = 0 \text{ si } x < 0$$

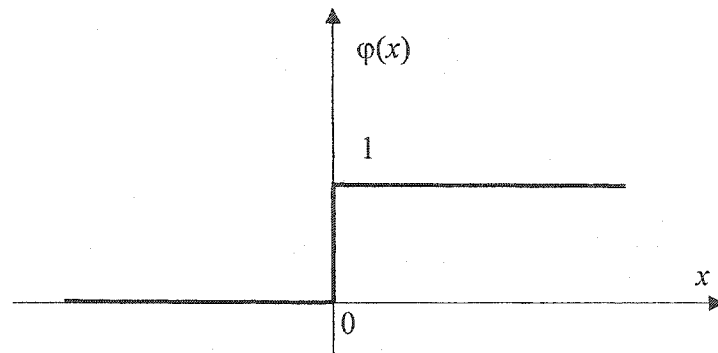
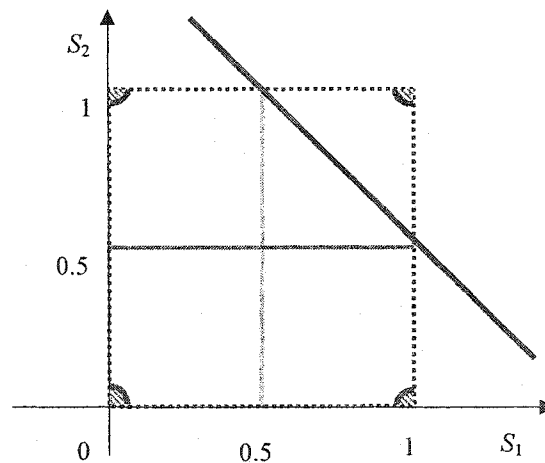


Figure 1.9 Fonction Heaviside

Dans ce cas, les grandeurs S_1 et S_2 ne peuvent plus prendre que les valeurs 0 et 1.

La densité des points images sera nulle partout, mis à part dans les quatre coins du carré.

Comme nous le voyons dans la figure ci-dessous, avec une seule droite il est possible de séparer les points images appartenant au carré supérieur droit (P^{++}) du reste des points images.

Figure 1.10 Espace caché pour a égale l'infini

1.5 Réseau à trois couches et trois neurones

Nous venons de voir que si a est égale à l'infini alors une seule droite est capable de séparer l'ensemble des points images de V_1 de ceux de V_2 . Pour tracer une droite dans le plan S_1S_2 il suffit d'un seul neurone. Nous utilisons les sorties S_1 et S_2 comme entrées pour ce nouveau neurone qui va donc former une nouvelle couche, et qui va permettre de tracer une droite d_3 dans le plan S_1S_2 . Cependant, dans la figure 1.8 une seule droite n'est pas suffisante pour isoler parfaitement le carré P'^+ du reste du carré. Mais étant donné le résultat que nous venons d'exposer avec la fonction Heaviside et dans le but de trouver un compromis entre la performance de notre réseau et sa simplicité, nous avons décidé de n'ajouter qu'un seul neurone. Nous obtenons le réseau à trois couches suivant:

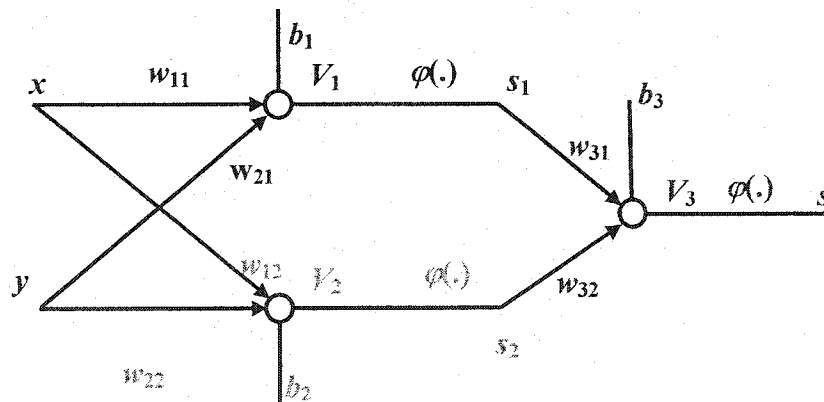


Figure 1.11 Réseau à deux entrées, une sortie et deux neurones cachés

Les équations du neurone de sortie sont:

$$V_3 = w_{31} S_1 + w_{32} S_2 + b_3 \quad 1.7$$

$$S = \varphi(V_3)$$

$$S = \frac{1}{1 + e^{-aV_3}} \quad 1.8$$

Voici quelques possibilités quant au positionnement de la droite d_3 , générée par le neurone de sortie, dont l'équation est :

$$w_{31}S_1 + w_{32}S_2 + b_3 = 0$$

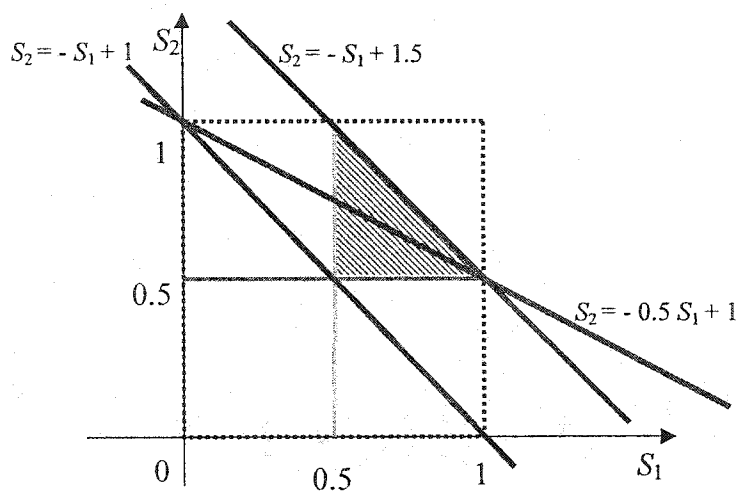


Figure 1.12 Différentes possibilités pour la droite d_3

1.5.1 Lemme 1

Soit un réseau de neurones dont l'architecture est donnée par la figure 1.11. Ce réseau est capable de séparer deux classes C_1 et C_2 , qui sont séparables par une frontière bilinéaire, si la fonction d'activation utilisée est la fonction Heaviside.

Preuve :

La figure 1.10 représente l'espace caché pour le réseau de la figure 1.11 lorsque nous utilisons la fonction Heaviside. Nous avons vu que dans ce cas une seule droite permet de séparer les points images appartenant au carré supérieur droit (P^+) du reste des points images, donc une seule droite est capable de séparer les points images de V_1 des points images de V_2 . Par conséquent, le réseau proposé dans la figure 1.11, permet de tracer une droite, d_3 , dans l'espace caché, qui sépare la classe C_1 de C_2 . En fait dans ce cas tous les points de P^+ seront séparés du reste du plan xy , et le réseau va reproduire la frontière bilinéaire.

En pratique, le paramètre a ne peut prendre que des valeurs finies. En effet les performances des algorithmes d'apprentissage d'un réseau dépendent de la valeur de ce paramètre. Par exemple, l'algorithme de rétro propagation ne peut pas être utilisé si a est égale à l'infini. Donc bien qu'avec la fonction Heaviside nous obtenons un résultat parfait, en pratique cette fonction est inutilisable puisque nous ne pouvons pas l'utiliser pour l'apprentissage du réseau. Nous devons alors étudier la fonction sigmoïde.

1.6 Étude pratique de la fonction sigmoïde

Nous savons que les réseaux de neurones sont principalement simulés à l'aide des ordinateurs. Donc, il est intéressant d'étudier l'effet de l'utilisation de l'informatique sur le réseau. L'algorithme de *Rétro propagation*, par exemple, est un outil informatique

dont les performances peuvent être affectées par les approximations des ordinateurs. Dans cette section nous allons faire une analyse pratique de quelques aspects que nous étudions.

En théorie, nous avons noté que la sigmoïde a deux asymptotes l'une en 0 et l'autre en 1. En pratique, cependant, il y a des approximations.

Avec MatLab :

$$1/(1+e^{-10}) = 1$$

$$1/(1+e^{710}) = 0$$

Avec MAPLE :

$$1/(1+e^{-21.5}) = 1$$

$$1/(1+e^{4944763831}) = 0$$

Comme les valeurs ci-dessus nous le prouvent, avec l'utilisation de l'ordinateur, les asymptotes n'existent pas. Voici d'ailleurs l'allure de la fonction sigmoïde avec les approximations. La figure 1.13 qui suit n'est pas exacte, elle est exagérée afin de bien visualiser l'effet des approximations

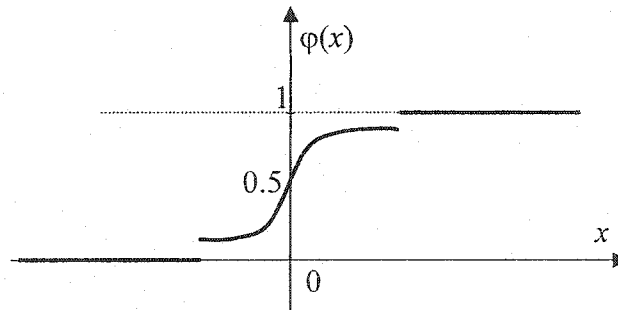


Figure 1.13 Allure de la fonction sigmoïde pratique

L'espace caché obtenu avec ces approximations est le même carré unitaire que celui de la figure 1.10, mais dans ce cas, les côtés sont inclus.

1.6.1 Proposition 1

Soit un réseau de neurones dont l'architecture est donnée par la figure 1.11 utilisant une sigmoïde comme fonction d'activation. Soit des données dont les coordonnées sont connues, et provenant de deux classes séparables par une frontière bilinéaire. Il est possible de trouver une infinité de valeurs finies pour le paramètre a de la sigmoïde qui va permettre au réseau de parfaitement séparer les deux classes.

Preuve :

Soit :

D l'ensemble des données.

N le nombre total de données contenues dans D .

N_1 l'ensemble des données de la classe C_1 .

N_2 l'ensemble des données de la classe C_2 .

d_i correspond à la $i^{\text{ème}}$ de ces données pour $i \in [1, N]$.

V_{1i}^+ correspond à la sortie linéaire du premier neurone pour $i \in N_1$.

V_{2i}^+ correspond à la sortie linéaire du deuxième neurone pour $i \in N_1$.

V_{1j}^- correspond à la sortie linéaire du premier neurone pour $j \in N_2$.

V_{2j}^- correspond à la sortie linéaire du deuxième neurone pour $j \in N_2$.

Nous avons déjà vu (p.22) les résultats suivants :

Avec MatLab, version 6.1.0.450, release 12.1:

$$1/(1+e^{-10}) = 1 \text{ et } 1/(1+e^{710}) = 0$$

Avec MAPLE, version 7.00 :

$$1/(1+e^{-21.5}) = 1 \text{ et } 1/(1+e^{4944763831}) = 0$$

Soient les données et la séparation bilinéaire que nous recherchons :

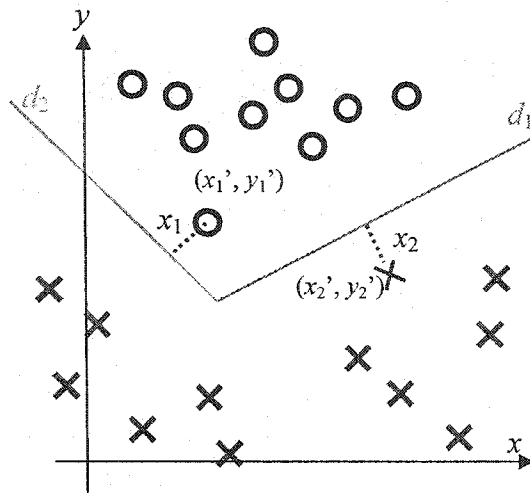


Figure 1.14 Données et surface de séparation

Soit (x_1', y_1') les coordonnées de la donnée la plus proche des deux droites et appartenant à son demi-plan défini positif, notons par x_1 la distance entre cette donnée et la droite la plus proche (Figure 1.14).

Soit (x_2', y_2') les coordonnées de la donnée la plus proche des deux droites et appartenant à son demi-plan défini négatif, notons par x_2 la distance entre cette donnée et la droite la plus proche (Figure 1.14).

Si la distance x_1 est relative à la droite

- d_1 , alors posons :

$$V^+ = w_{11}x_1' + y_1' + b_1$$

- d_2 , (Figure 1.14) alors posons :

$$V^+ = w_{21}x_1' + y_1' + b_2$$

V^+ est donc la plus petite valeur positive que peut prendre V_1 ou V_2 .

Si x_1 , est défini par rapport à n'importe quelle droite (donc si la donnée est équidistante des deux droites), alors nous pouvons définir V^+ avec n'importe laquelle des deux équations présentées plus haut.

Si la distance x_2 est relative à la droite

- d_1 , (Figure 1.14) alors posons :

$$V^+ = w_{11}x_2' + w_{12}y_2' + b_1$$

- d_2 , alors posons :

$$V^+ = w_{21}x_2' + w_{22}y_2' + b_2$$

V^+ est la plus petite valeur négative possible que peut prendre V_1 ou V_2 .

Si x_2 , est défini par rapport à n'importe quelle droite (donc si la donnée est équidistante des deux droites), alors nous pouvons définir V^+ avec n'importe laquelle des deux équations présentées plus haut.

Nous voulons déterminer la valeur de a à partir de laquelle l'image de chacune de nos données sera à l'un des coins du carré. Pour ce faire, il faut déterminer la valeur de a pour laquelle :

$$\left\{ \begin{array}{ll} \frac{1}{1 + e^{[-aV_{1i}^+]}} = 1, & \frac{1}{1 + e^{[-aV_{2i}^+]}} = 1, \quad \forall i \in N \\ \frac{1}{1 + e^{[-aV_{1j}^-]}} = 0, & \frac{1}{1 + e^{[-aV_{2j}^-]}} = 0, \quad \forall j \in N \end{array} \right. \quad 1.9$$

Puisque que V^+ est la plus petite valeur positive pour V_1 et V_2 et V^- est la plus petite valeur négative possible pour V_1 et V_2 . Donc, le système 1.11 est vérifié si :

$$\frac{1}{1 + e^{[-aV^+]}} = 1$$

Et

$$\frac{1}{1 + e^{[-aV^-]}} = 0$$

Si nous travaillons avec MatLab,

Nous savons que

$$V^+ > 0 \Rightarrow a V^+ > 0$$

Donc pour que

$$1/(1 + \exp(-a V^+)) = 1$$

il faut que

$$-a (V^+) \leq -10$$

$$\Rightarrow a \geq (10/V^+)$$

de plus

$$V^- < 0 \Rightarrow a V^- < 0.$$

Donc pour que

$$1/(1 + \exp(-aV^-)) = 0$$

il faut que

$$-a (V^-) \leq -710$$

$$\Rightarrow a \geq (-710/V^-)$$

donc

$$a = \max \{(10/V^{++}), (-710/V^-)\} \quad 1.10$$

Cette valeur de a est suffisamment grande pour que l'image de chacune des données se situe à l'un des coins du carré.

Avec MAPLE et en procédant de la même manière, nous obtenons :

$$a = \max \{(21.5/V^{++}), (-4944763831/V^-)\} \quad 1.11$$

En utilisant cette méthode pour fixer a , voici l'espace caché que nous obtenons (figure 1.15).

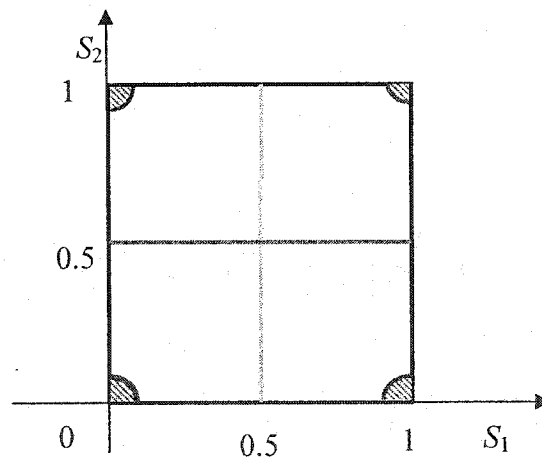


Figure 1.15 Séparation parfaite de données

Cette figure a des caractéristiques identiques à celles de la figure 1.10, en effet les deux représentent un espace caché où tous les points images sont concentrés dans les coins du carré, et une seule droite permet de séparer toutes les données des deux classes.

Les équations 1.10 et 1.11 donnent la valeur minimum de a qui va permettre d'obtenir

un réseau qui sépare parfaitement les deux classes. Toutes les valeurs supérieures à ce minimum sont aussi valides.

Note :

Si une donnée appartient à la droite d_1 (dans la partie de cette droite qui ne forme pas la frontière que nous étudions), alors quelque soit la valeur de a , l'image de cette donnée va être sur le segment de droite d'_1 . De même si une donnée appartient à la demi-droite d_2 alors son image va être sur le segment d'_2 . Mais avec le raisonnement que nous venons de faire il nous sera toujours possible de trouver une droite qui va faire une parfaite séparation de deux classes.

Notons que l'étude que nous venons de faire est basée sur des approximations dues à l'utilisation de l'ordinateur, cette étude ne peut pas être utilisée pour une étude théorique. Nous venons donc de faire une étude pratique qui a permis de montrer qu'il existe une valeur finie du paramètre a à partir de laquelle il y a une séparation parfaite des classes C_1 et C_2 . Cependant placer tous les points images dans les coins du carré est une condition trop restrictive ; en effet nous allons voir comment nous pouvons changer cette condition en une autre moins stricte. Ceci nous permettra d'obtenir un résultat théorique plus intéressant. Dans la section suivante, nous allons donc nous concentrer sur l'aspect théorique de la séparation faite par d_3 .

1.7 Théorème 1

Soit un réseau de neurones dont l'architecture est donnée par la figure 1.11 utilisant une sigmoïde comme fonction d'activation. Soit des données dont les coordonnées sont connues, et provenant de deux classes séparables par une frontière bilinéaire. Alors il existe une valeur pour le paramètre a , de la sigmoïde, à partir de laquelle le réseau sera capable de parfaitement isoler les deux classes.

Preuve :

Nous savons que les demi-droites séparent le plan xy en quatre parties que voici :

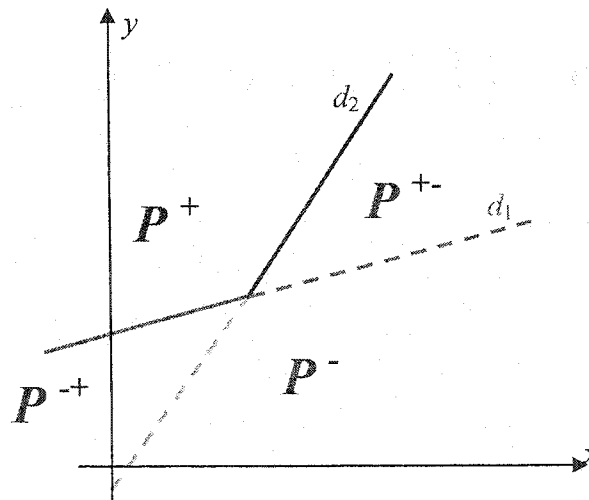


Figure 1.16 Deux demi-plans positifs et deux autres négatifs

Chacune des deux droites engendre un demi-plan positif et un autre négatif.

Soit,

- P^+ l'intersection des deux demi-plans positifs (Figure 1.16)

- P^- l'intersection des deux demi-plans négatifs
- P^{+-} l'intersection du demi-plan positif de d_1 avec le demi-plan négatif défini par d_2
- P^{+} l'intersection du demi-plan négatif de d_1 avec le demi-plan positif défini par d_2
- C_1 la classe qui appartient à P^+

Nous fixons une certaine valeur positive pour a , avec cette valeur nous plaçons dans le plan S_1S_2 les points images de toutes les données. De ces points images, nous devons en choisir deux (P_1 et P_2), appartenant à la classe C_2 , qui forment une droite d_3 , tel que tous les points de la classe C_2 sont du même côté de cette droite (Figure 1.17). De plus le point $(0, 0)$ doit être dans le demi-plan négatif de cette droite à pente négative. Par exemple, dans la figure 1.17 ce sont les deux croix encadrées qui définissent P_1 et P_2 , en effet, toutes les données appartenant à C_2 (x) sont d'un même côté de cette droite.

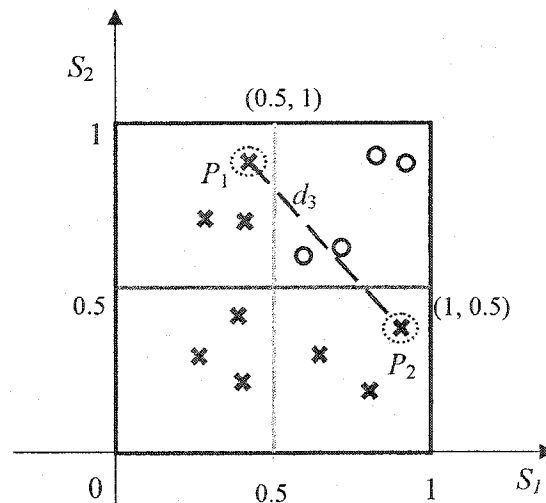


Figure 1.17 Espace caché et recherche du a minimum

Si nous augmentons graduellement la valeur de a alors :

- Les points images des données contenues dans le plan P^- se dirigent vers le point $(0, 0)$
- Les points images des données contenues dans le plan P^{+-} se dirigent vers le point $(1, 0)$
- Les points images des données contenues dans le plan P^{+} se dirigent vers le point $(0, 1)$
- Les points images des données contenues dans le plan P^{++} se dirigent vers le point $(1, 1)$

Or nous savons, à l'aide du lemme 1, qu'à la limite si a est égale à l'infini alors les points images vont se trouver dans les coins, et la droite d_3 va être capable de séparer les deux classes. Nous pouvons dire maintenant qu'il existe une valeur de a dans l'intervalle $]0, \infty]$ qui permet de parfaitement séparer les deux classes.

Nous venons de montrer qu'il existe une valeur de a à partir de laquelle nous pouvons réaliser la classification désirée. Afin de trouver cette valeur, qui est la valeur minimum, nous devons fixer une valeur pour a , tracer les points images et tracer la droite d_3 suivant la méthode expliquée.

Si avec cette valeur de a la séparation est bonne alors il faut répéter avec une valeur plus petite pour a et retracer la droite d_3 et vérifier la validité de la séparation. La valeur minimum recherchée est la plus petite possible qui va bien séparer les données.

Si avec cette valeur de a la séparation est mauvaise alors il faut répéter avec une valeur

plus grande pour a et retracer la droite d_3 et vérifier la validité de la séparation. La valeur minimum recherchée est la plus petite possible qui va bien séparer les données.

Une fois la valeur minimum de a obtenue, nous utilisons l'équation de la droite formée par les deux points P_1 et P_2 afin de fixer les paramètres du neurone de sortie.

Voici un exemple dans lequel nous fixons les paramètres du neurone de sortie à partir des points P_1 et P_2 .

Si l'équation de la droite formée par P_1 et P_2 est :

$$2S_1 + 3S_2 + 1 = 0$$

Alors par analogie avec l'équation

$$w_{31}S_1 + w_{32}S_2 + b_3 = 0$$

Nous devons fixer :

$$w_{31} = 2 \quad w_{32} = 3 \quad \text{et} \quad b_3 = 1.$$

L'exemple qui suit va nous permettre de bien comprendre la méthode que nous venons d'exposer.

Exemple 1.1

Voici un exemple dans lequel nous cherchons la valeur minimum de a et les valeurs des paramètres du neurone de sortie correspondantes.

Les équations des deux droites formant la frontière sont :

$$y = -2/5 x + 4.2$$

$$y = 3/5 x - 3.2$$

Voici les coordonnées des données à séparer

(5, 2.5), (6, 1.81), (7, 0.5), (5, 2.1), (7.2, 1.25), (7.5, 1.5), (8, 1.3), (7.4, 1.2), (9, 3.5), (9, 4), (10, 2) et (9, 2).

Voici l'espace des entrées (figure 1.18).

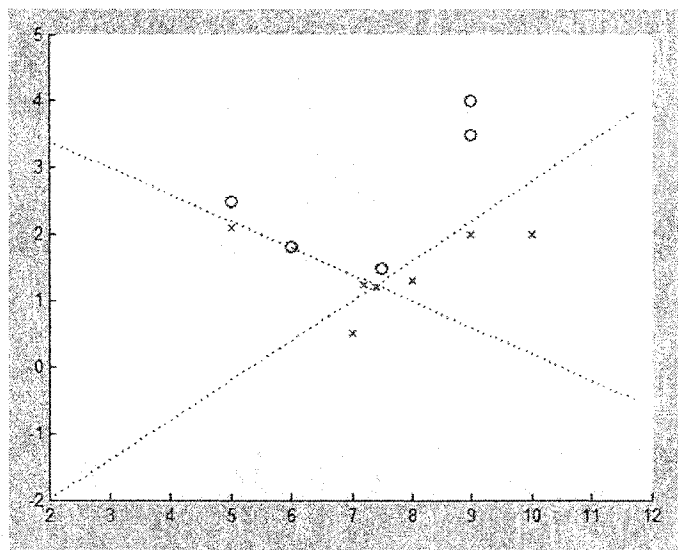


Figure 1.18 Différentes données dans l'espace des entrées

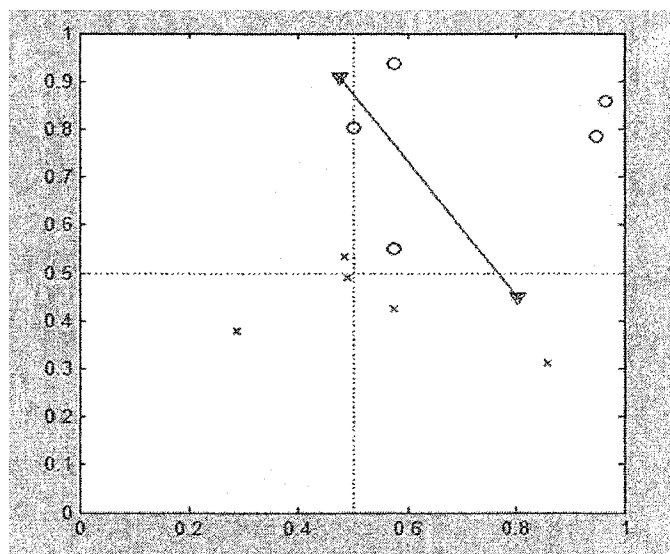


Figure 1.19 Espace caché pour $a = 1$

L'espace caché pour $a = 1$ est représenté ci-dessus, figure 1.19. Les deux croix entourées d'un triangle représentent les deux points P_1 et P_2 qui vont permettre de tracer la droite de séparation (d_3). Après avoir tracé d_3 , nous remarquons que la séparation des deux classes n'est pas parfaite : deux données représentées par des cercles sont mal classées, elles sont dans le mauvais demi-plan relativement à la droite.

Nous avons graduellement augmenté la valeur de a jusqu'à 3.3.

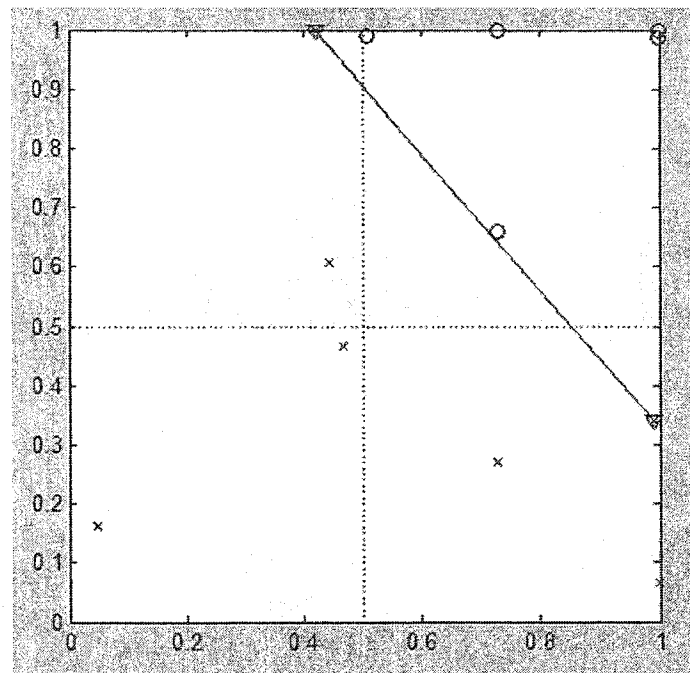


Figure 1.20 Espace caché pour $a = 3.3$

Dans la figure 1.20, nous voyons que la droite sépare parfaitement les deux classes. Sachant que pour $a = 3.2$ la séparation n'est pas parfaite, nous pouvons affirmer que $a = 3.3$ est la valeur minimum pour laquelle le réseau proposé est capable de faire une

séparation parfaite des deux classes. Bien sûr, il est possible de trouver une valeur plus précise avec plus de chiffres significatifs. Les coordonnées des deux points formant d_3 sont : (0.9902, 0.3497) et (0.4182, 0.9995). Plus loin dans ce rapport nous reviendrons sur cet exemple.

Nous avons montré, qu'avec un réseau à trois couches, si nous avons nos données et la frontière bilinéaire, il est possible de trouver une valeur pour a qui va permettre de parfaitement séparer les deux classes. Cependant, séparer deux classes n'est pas la même chose que de reproduire une frontière bilinéaire. Dans le cas où notre but serait de reconstruire la frontière bilinéaire, il faudrait que a tende vers l'infini, or comme nous l'avons déjà mentionné, cette valeur ne peut pas être utilisée en pratique.

Dans ce premier chapitre, nous avons montré comment, en faisant varier la valeur du paramètre de la sigmoïde, il est possible d'obtenir une parfaite séparation de deux classes séparables par une frontière bilinéaire. Dans le chapitre suivant, nous allons nous concentrer sur la capacité du réseau à approximer une frontière bilinéaire.

CHAPITRE 2 ÉQUATION EXPLICITE

Dans ce chapitre, nous allons obtenir une fonction qui représente la frontière engendrée par le réseau. Cette fonction explicite de la forme $y = f(x)$ permettra d'analyser l'effet de chaque paramètre du réseau (poids, biais...) sur la frontière qu'il génère.

2.1 Équation explicite de la frontière

2.1.1 Lemme 2

Soit un réseau dont l'architecture est donnée par la figure 1.11 et dont la fonction d'activation est la sigmoïde. Nous pouvons obtenir une équation explicite de la frontière engendrée par le réseau s'il y a deux neurones dans la couche cachée, et que les deux poids issus de l'une des entrées sont unitaires.

Preuve :

Dans le chapitre précédent, nous avons vu comment, à partir des équations de deux droites, obtenir les équations des sorties linéaires du réseau montré dans la figure 1.11 (Équation 1.4).

$$V_1 = w_{11}x + y + b_1$$

$$V_2 = w_{21}x + y + b_2$$

La restriction dans le lemme 2 (les deux poids issus de l'une des entrées sont unitaires) se traduit par le fait que nous n'avons pas de poids associés à y . Ceci est dû au fait que nous avons utilisé des droites de la forme

$$y = ax + b$$

et non pas de la forme

$$ax + by + c = 0$$

Remarque :

Nous venons de voir que notre étude va se faire sur un réseau ayant un poids fixe sur chacun de ses deux neurones cachés. Cette restriction semble affecter l'étude du réseau. Cependant, il a été démontré (Labib et Assadi, 2004) qu'il est possible de fixer un des poids de chacun des neurones cachés d'un MLP sans affecter la performance générale du réseau. Donc, le fait que les poids issus de l'une des entrées sont unitaires n'affecte pas l'étude que nous faisons.

Nous pouvons écrire,

$$S_1 = \frac{1}{1 + e^{-a(w_{11}x + y + b_1)}}$$

$$S_2 = \frac{1}{1 + e^{-a(w_{21}x + y + b_2)}}$$

De plus,

$$V_3 = w_{31}S_1 + w_{32}S_2 + b_3$$

$$\Rightarrow V_3 = w_{31} \frac{1}{1 + e^{-a(w_{11}x + y + b_1)}} + w_{32} \frac{1}{1 + e^{-a(w_{21}x + y + b_2)}} + b_3 \quad 2.1$$

Soit

$$A = w_{11}x + b_1$$

$$B = w_{21}x + b_2$$

Sur la frontière du réseau, la sortie V_3 est nulle ($V_3 = 0$), l'équation 2.1 devient :

$$w_{31} \frac{1}{1 + e^{-a(A+y)}} + w_{32} \frac{1}{1 + e^{-a(B+y)}} + b_3 = 0$$

En fixant

$$C = e^{-aA}$$

$$D = e^{-aB}$$

$$\Rightarrow w_{31} \frac{1}{1 + Ce^{-ay}} + w_{32} \frac{1}{1 + De^{-ay}} + b_3 = 0$$

$$\Rightarrow \frac{w_{31}(1 + De^{-ay}) + w_{32}(1 + Ce^{-ay})}{(1 + Ce^{-ay})(1 + De^{-ay})} + b_3 = 0$$

$$\Rightarrow w_{31} + w_{31}De^{-ay} + w_{32} + w_{32}Ce^{-ay} + b_3[1 + Ce^{-ay}][1 + De^{-ay}] = 0 \quad 2.2$$

Effectuons les substitutions suivantes :

$$X = e^{(-y)}$$

$$\left. \begin{aligned} T_1 &= b_3CD \text{ donc } T_1 \neq 0 \text{ car } (C, D, b_3) \neq 0 \\ T_2 &= w_{31}D + b_3D w_{31} + w_{32}C + b_3C \\ T_3 &= b_3 + w_{32} + w_{31} \end{aligned} \right\} \quad 2.3$$

L'équation 2.2 devient :

$$T_1 (X)^{2a} + T_2 (X)^a + T_3 = 0$$

Avec une dernière substitution,

$$Y = (X)^a$$

Nous obtenons:

$$T_1 Y^2 + T_2 Y + T_3 = 0 \quad 2.4$$

C'est une équation du second degré ($T_1 \neq 0$) que nous pouvons résoudre.

Les deux solutions de cette équation sont :

$$Y = \frac{-T_2 \pm \sqrt{T_2^2 - 4T_1 T_3}}{2T_1}$$

Donc

$$(X)^a = \frac{-T_2 + \sqrt{T_2^2 - 4T_1 T_3}}{2T_1} \Rightarrow X = \sqrt[a]{\frac{-T_2 + \sqrt{T_2^2 - 4T_1 T_3}}{2T_1}}$$

Ou

$$(X)^a = \frac{-T_2 - \sqrt{T_2^2 - 4T_1 T_3}}{2T_1} \Rightarrow X = \sqrt[a]{\frac{-T_2 - \sqrt{T_2^2 - 4T_1 T_3}}{2T_1}}$$

De plus, nous avons :

$$X = e^{-y} \text{ avec } X > 0$$

Donc

$$X = e^{-y} = \sqrt[a]{\frac{-T_2 \pm \sqrt{T_2^2 - 4T_1 T_3}}{2T_1}}$$

$$\Rightarrow \ln(e^{-y}) = \ln \left(\sqrt[3]{\frac{-T_2 \pm \sqrt{T_2^2 - 4T_1T_3}}{2T_1}} \right)$$

$$\Rightarrow y = -\ln \left(\sqrt[3]{\frac{-T_2 \pm \sqrt{T_2^2 - 4T_1T_3}}{2T_1}} \right)$$

En substituant les variables T_1 , T_2 et T_3 par leurs valeurs en fonction de C et D nous obtenons l'équation suivante :

$$y = -\ln \left(\sqrt[3]{\frac{-(w_{31}D + b_3D + w_{32}C + b_3C) \pm \sqrt{(w_{31}D + b_3D + w_{32}C + b_3C)^2 - 4b_3CD(b_3 + w_{32} + w_{31})}}{2b_3CD}} \right)$$

ou

$$y = -\frac{1}{a} \ln \left(\frac{\frac{-(w_{31} + b_3)e^{-aB} + (w_{32} + b_3)e^{-aA}}{2b_3e^{-aA}e^{-aB}} \pm \sqrt{\left[\frac{-(w_{31} + b_3)e^{-aB} + (w_{32} + b_3)e^{-aA}}{2b_3e^{-aA}e^{-aB}} \right]^2 - 4b_3e^{-aA}e^{-aB}((b_3 + w_{32} + w_{31}))}}{2b_3e^{-aA}e^{-aB}} \right)$$

Finalement, voici y en fonction des paramètres du réseau :

$$y = -\frac{1}{a} \ln \left(\frac{\frac{-(w_{31} + b_3)e^{-a(w_{21}x + b_2)} + (w_{32} + b_3)e^{-a(w_{11}x + b_1)}}{2b_3e^{-a(w_{11}x + b_1)}e^{-a(w_{21}x + b_2)}} \pm \sqrt{\left[\frac{-(w_{31} + b_3)e^{-a(w_{21}x + b_2)} + (w_{32} + b_3)e^{-a(w_{11}x + b_1)}}{2b_3e^{-a(w_{11}x + b_1)}e^{-a(w_{21}x + b_2)}} \right]^2 - 4b_3e^{-a(w_{11}x + b_1)}e^{-a(w_{21}x + b_2)}((b_3 + w_{32} + w_{31}))}}{2b_3e^{-a(w_{11}x + b_1)}e^{-a(w_{21}x + b_2)}} \right) \quad 2.5$$

Signe de la racine carrée

Dans l'équation 2.5, il existe deux possibilités pour le signe de la racine carrée. Chacune représente une des deux solutions de l'équation du second degré (Équation 2.4). Faisons

une analyse afin de savoir quel est le signe que nous devons employer. Le terme à l'intérieur du logarithme doit être positif pour que l'équation soit réelle. Or, dans le premier chapitre, nous avons ajouté un troisième neurone pour tracer la droite d_3 (Figure 1.12), cette droite doit être de pente négative, et le point (1, 1) devrait se retrouver dans son demi-plan positif. Ces deux constatations peuvent se traduire par les inégalités mathématiques suivantes :

- $w_{31}x + w_{32}y + b_3 > 0$ pour $x = 1$ et $y = 1$
 $\Rightarrow w_{31} + w_{32} + b_3 > 0 \Rightarrow T_3 > 0$
- $b_3 < 0 \Rightarrow -4b_3 e^{[-a(w_{11}x+b_1)]} e^{[-a(w_{21}x+b_2)]} > 0$
 $\Rightarrow -4T_1 > 0$
 $\Rightarrow -4T_1T_3 > 0$
- $[(w_{31} + b_3) e^{[-a(w_{21}x+b_2)]} + (w_{32} + b_3) e^{[-a(w_{11}x+b_1)]}]^2 > 0$

Nous savons que

$$\sqrt{T_2^2} \geq T_2$$

Or nous venons de montrer que :

$$-4T_1T_3 > 0$$

Donc

$$\begin{aligned} \sqrt{T_2^2 - 4T_1T_3} &> T_2 \\ \Rightarrow \sqrt{T_2^2 - 4T_1T_3} - T_2 &> 0 \end{aligned}$$

De plus

$$T_2^2 \text{ et } -4T_1T_3 > 0$$

Donc

$$\left| \sqrt{T_2^2 - 4T_1T_3} \right| > |T_2|$$

Par suite

$$-\sqrt{T_2^2 - 4T_1T_3} - T_2 < 0$$

De plus

$$b_3 < 0 \Rightarrow 2T_1 < 0$$

Donc

$$\frac{-T_2 + \sqrt{T_2^2 - 4T_1T_3}}{2T_1} < 0$$

Et

$$\frac{-T_2 - \sqrt{T_2^2 - 4T_1T_3}}{2T_1} > 0$$

Donc, pour obtenir une valeur réelle pour y , nous devons utiliser la forme suivante de l'équation :

$$y = -\ln \left(\sqrt[4]{\frac{-T_2 - \sqrt{T_2^2 - 4T_1T_3}}{2T_1}} \right)$$

En fonction des paramètres initiaux nous obtenons :

$$y = -\frac{1}{a} \ln \left(\frac{-\left[(w_{31} + b_3) e^{-a(w_{21}x + b_2)} \right] + (w_{32} + b_3) e^{-a(w_{11}x + b_1)}}{2b_3 e^{-a(w_{11}x + b_1)} e^{-a(w_{21}x + b_2)}} \right) \sqrt{\frac{\left[(w_{31} + b_3) e^{-a(w_{21}x + b_2)} \right] + (w_{32} + b_3) e^{-a(w_{11}x + b_1)}}{2b_3 e^{-a(w_{11}x + b_1)} e^{-a(w_{21}x + b_2)}}}^2 - 4b_3 e^{-a(w_{11}x + b_1)} e^{-a(w_{21}x + b_2)} ((b_3 + w_{32} + w_{31}))} \right) \quad 2.6$$

Cette équation 2.6 représente l'équation explicite de la forme $y = f(x)$, elle permet de tracer la courbe générée par le réseau.

En guise d'exemple, nous allons utiliser MAPLE afin de tracer cette fonction.

L'étude que nous faisons se fait dans un intervalle fini et non pas sur la totalité du plan.

Ainsi nous devons définir un espace de variation des données.

Exemple 2.1:

$$w_1 = -3/5, b_1 = 3.2$$

$$w_2 = 2/5, b_2 = -4.2$$

$$w_{31}=1; w_{32} = 1, b_3 = -1.4; a = 2$$

L'espace de variation est le suivant :

$$x \in [8.5, 13.5] \text{ et } y \in [0, 2.7]$$

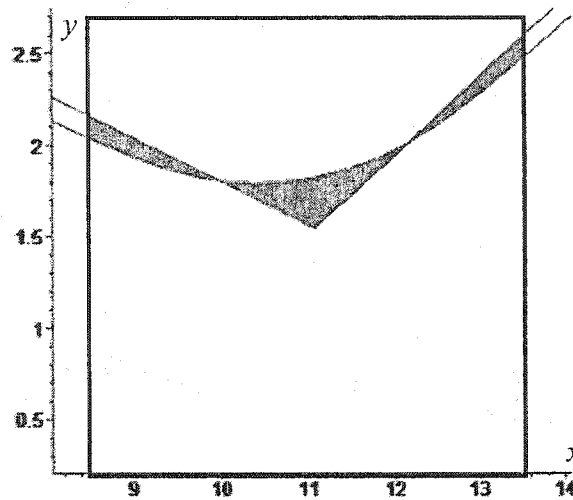


Figure 2.1 Courbe générée par le réseau

Dans la figure 2.1, nous voyons la frontière bilinéaire à reproduire, c'est celle qui est formée par les deux demi-droites et la surface de variation est celle délimitée par le rectangle. Les surfaces pleines représentent la surface d'erreur qui est proportionnelle au pourcentage d'erreur. Nous verrons plus loin en détail ce que représente le pourcentage d'erreur.

Dans cet exemple, quand nous nous éloignons de la discontinuité, point d'intersection des deux demi-droites, la courbe semble bien suivre la frontière.

Notons que dans l'exemple illustré il y a deux points d'intersection entre la courbe et les droites, ceci n'est pas toujours le cas. Justement, nous allons maintenant trouver des équations analytiques qui nous permettront de savoir s'il existe aucun, un ou deux points d'intersection. Nous allons aussi trouver des équations qui permettent de trouver les coordonnées des intersections si elles existent. La surface d'erreur dans la figure 2.1

varie en fonction des intersections, nous avons donc besoin de leurs coordonnées afin de pouvoir exprimer cette surface. Voici une des raisons qui nous motivent à chercher ces équations.

2.1.2 Intersection entre les droites et la surface de séparation

Trouvons, s'il existe, un point d'intersection entre la droite d_1 et la surface de séparation :

Sur la droite d_1

$$w_{11}x + y + b_1 = 0$$

$$\Rightarrow \exp [-a (w_{11}x + y + b_1)] = \exp(0) = 1$$

Donc, pour les points qui appartiennent à la droite d_1 , l'équation 2.1 se simplifie et devient:

$$V_3 = w_{31} \frac{1}{1+1} + w_{32} \frac{1}{1 + e^{[-a(w_{21}x + y + b_2)]}} + b_3 \quad 2.7$$

Nous cherchons un point qui appartient aussi à la frontière tracée par le réseau, donc un point où $V_3 = 0$.

Nous pouvons donc réécrire l'équation 2.7 de cette façon:

$$\begin{aligned} 0 &= w_{31} \frac{1}{2} + w_{32} \frac{1}{1 + e^{[-a(w_{21}x + y + b_2)]}} + b_3 \\ 0 &= w_{31} \{1 + e^{[-a(w_{21}x + y + b_2)]}\} + 2w_{32} + 2b_3 \{1 + e^{[-a(w_{21}x + y + b_2)]}\} \\ \Rightarrow \{w_{31} + 2w_{32} + 2b_3\} + e^{[-a(w_{21}x + y + b_2)]} [w_{31} + 2b_3] &= 0 \end{aligned}$$

En utilisant l'équation ci-dessus et l'équation de la droite, nous obtenons le système suivant :

$$\begin{cases} w_{11}x + y + b_1 = 0 \\ \{w_{31} + 2w_{32} + 2b_3\} + e^{[-a(w_{21}x + y + b_2)]}[w_{31} + 2b_3] = 0 \end{cases} \quad 2.8$$

Nous avons donc un système à deux équations et deux inconnues (x et y).

La solution de ce système existe, donc d_1 et la courbe ont un point d'intersection, si :

$$\begin{aligned} \left(-\frac{w_{31} + 2w_{32} + 2b_3}{w_{31} + 2b_3} \right) &> 0 \\ w_{31} &\neq 2b_3 \end{aligned} \quad 2.9$$

Le vecteur solution est donnée par :

$$\begin{cases} x = \frac{b_1 - b_2}{w_{21} - w_{11}} - \frac{1}{a(w_{21} - w_{11})} \ln \left(-\frac{w_{31} + 2w_{32} + 2b_3}{w_{31} + 2b_3} \right) \\ y = -w_{11}x - b_1 \end{cases} \quad 2.10$$

De la même façon, pour trouver le point d'intersection entre la courbe et la droite d_2 nous devons résoudre le système 2.9 suivant:

$$\begin{cases} w_{21}x + y + b_2 = 0 \\ \{w_{32} + 2w_{31} + 2b_3\} + e^{[-a(w_{11}x + y + b_1)]}[w_{32} + 2b_3] = 0 \end{cases} \quad 2.11$$

La solution de ce système existe si :

$$\left(-\frac{w_{32} + 2w_{31} + 2b_3}{w_{32} + 2b_3} \right) > 0 \quad 2.12$$

$$w_{32} \neq 2b_3$$

Le vecteur solution est donné par :

$$\begin{cases} x = \frac{b_2 - b_1}{w_{11} - w_{21}} - \frac{1}{a(w_{11} - w_{21})} \ln \left(-\frac{w_{32} + 2w_{31} + 2b_3}{w_{32} + 2b_3} \right) \\ y = -w_{21}x - b_2 \end{cases} \quad 2.13$$

Exemple 2.2:

Supposons deux droites d'équations :

$$-(3/5)x + y + 3.2 = 0$$

$$(2/5)x + y - 4.2 = 0$$

Avec ces deux équations, nous fixons les paramètres de la première couche du réseau de la façon suivante :

$$w_{11} = -3/5, b_1 = 3.2$$

$$w_{21} = 2/5, b_2 = -4.2$$

La valeur du paramètre a de la sigmoïde est 2.

Étudions différents cas possibles pour le neurone de sortie.

1^{er} cas :

$$w_{31} = 1; w_{32} = 1; b_3 = -1.6.$$

Voici la frontière dans l'espace caché (figure 2.2):

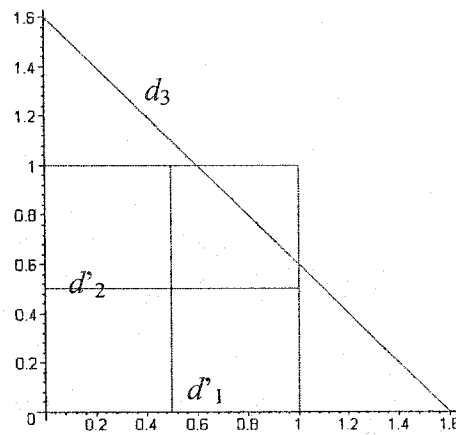


Figure 2.2 Droite d_3 sans intersection avec d'_1 et d'_2

Note :

- La droite d_3 ne coupe aucun des deux segments d'_1 et d'_2 .

Voici la frontière générée par le réseau (figure 2.3).

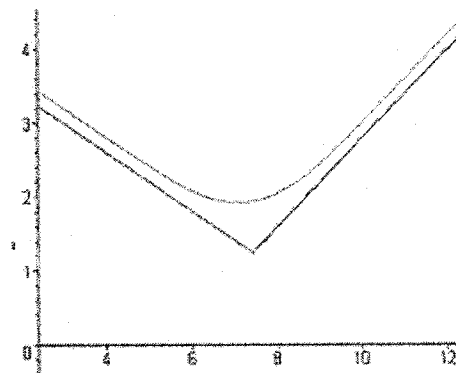


Figure 2.3 Exemple de frontière sans intersection

Graphiquement, nous constatons ici qu'il n'y a pas de points d'intersection.

Utilisons les inégalités 2.9 et 2.12 pour valider nos résultats.

$$\left(-\frac{w_{31} + 2w_{32} + 2b_3}{w_{31} + 2b_3} \right) = \left(-\frac{1 + 2 - 3.2}{1 - 3.2} \right) = \left(-\frac{-0.2}{-2.2} \right) < 0$$

De même

$$\left(-\frac{w_{32} + 2w_{31} + 2b_3}{w_{32} + 2b_3} \right) = \left(-\frac{-0.2}{-2.2} \right) < 0$$

Nous voyons donc que les deux inégalités ne sont pas vérifiées et les résultats analytiques confirment le fait que la courbe ne coupe ni la droite d_1 ni la droite d_2 .

2^{ème} cas :

$$w_{31} = 1; w_{32} = 1; b_3 = -1.4.$$

Voici la frontière dans l'espace caché (figure 2.4):

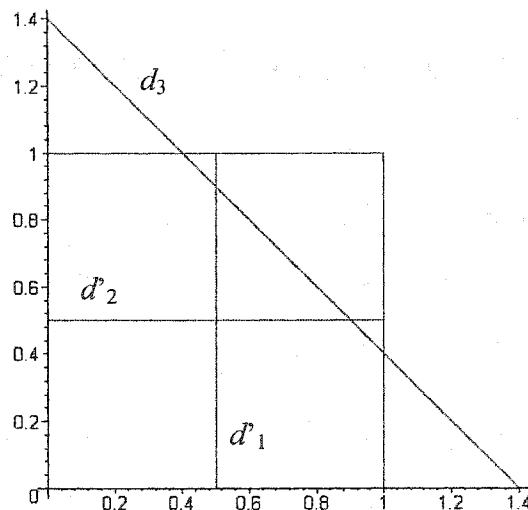


Figure 2.4 Droite d_3 avec deux intersections

Note :

- La droite d_3 coupe les deux segments d_1 et d_2 .

Voici la courbe obtenue avec MAPLE

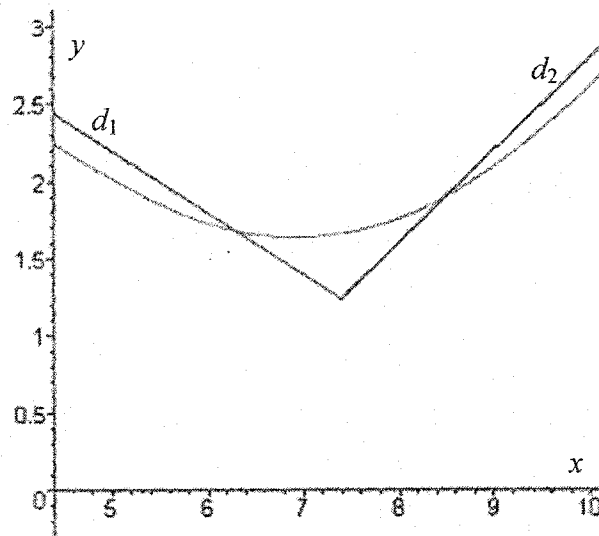


Figure 2.5 Exemple de frontière avec deux intersections

Cette figure 2.5 permet de voir deux points d'intersection.

Nous allons faire l'analyse analytique pour la droite d_2 .

À l'aide de l'inégalité 2.12 nous vérifions qu'il existe une intersection entre d_2 et la courbe, en effet :

$$\left(-\frac{w_{32} + 2w_{31} + 2b_3}{w_{32} + 2b_3} \right) = \left(-\frac{1 + 2 - 2.8}{1 - 2.8} \right) = \left(-\frac{0.2}{-1.8} \right) > 0$$

De plus

$$w_{32} \neq b_3$$

Les coordonnées du point d'intersection sont données par le système 2.13.

$$\begin{aligned}
 x &= \frac{b_2 - b_1}{w_{11} - w_{21}} - \frac{1}{a(w_{11} - w_{21})} \ln \left(-\frac{w_{32} + 2w_{31} + 2b_3}{w_{32} + 2b_3} \right) \\
 &= \frac{-4.2 - 3.2}{-\frac{3}{5} - \frac{2}{5}} - \frac{1}{2 \left(-\frac{3}{5} - \frac{2}{5} \right)} \ln \left(-\frac{0.2}{-1.8} \right) \\
 &= 6.301
 \end{aligned}$$

Et

$$y = 6.301(3/5) - 3.2$$

$$y = 1.679$$

Voici donc les coordonnées du point d'intersection : (6.301, 1.679). À l'aide de la figure 2.5, nous remarquons que le point d'intersection entre la courbe et la droite d_2 (pente négative) se situe effectivement dans les environs des coordonnées suivantes (6.301, 1.679).

De la même façon, nous pouvons prouver l'existence du point d'intersection entre la courbe et la droite d_1 et calculer ses coordonnées.

Sachant que les images des deux droites d_1 et d_2 sont les deux segments d'_1 et d'_2 et à l'aide des résultats que nous venons d'exposer, nous sommes capable de faire les constatations suivantes.

Remarques

Si la droite d_3 a :

- Un point d'intersection avec chacun des deux segments (d'_1 et d'_2), alors la courbe a un point d'intersection avec chacune des deux droites (d_1 et d_2).
- Un point d'intersection avec le segment d'_1 ou d'_2 , alors la courbe aura un point d'intersection avec la droite d_1 ou respectivement d_2 .

De plus,

- Si la droite d_3 n'a aucun point d'intersection avec les deux segments, alors la courbe n'aura pas de points d'intersection avec les droites d_1 et d_2 .
- Il existe au maximum un point d'intersection entre la frontière générée par le réseau et chacune des deux demi-droites. En effet, le segment d_3 ne peut couper qu'une seule fois chacun des deux segments d'_1 et d'_2 .

Dans les trois exemples qui suivent, obtenues à l'aide de l'équation 2.6 et de MAPLE, les demi-droites formant la frontière bilinéaire sont les mêmes, et leurs équations sont :

$$(-2/7)x + y + 1.3 = 0$$

$$(3/7)x + y - 7.4 = 0$$

Les poids et les biais des neurones cachés sont donc :

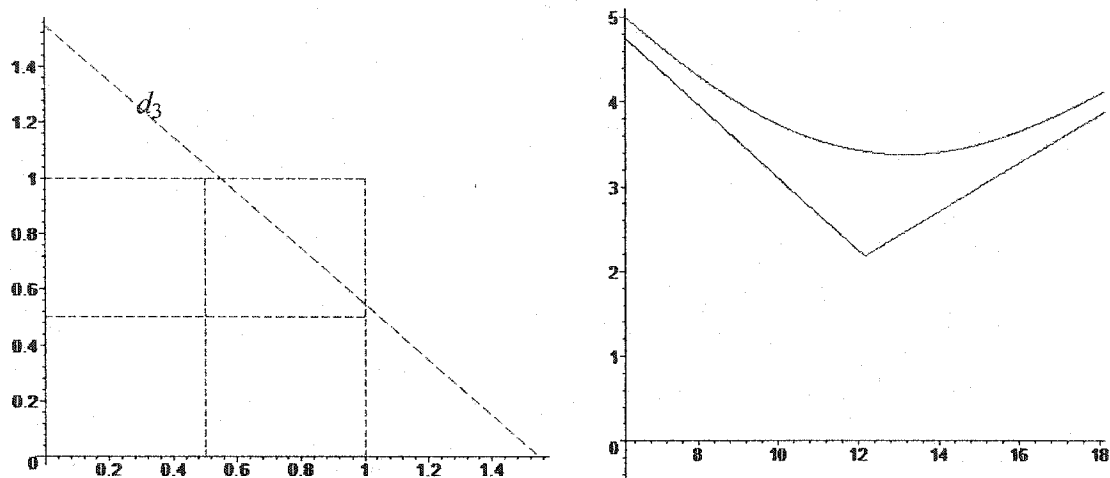
$$w_{11} = -2/7; b_1 = 1.3$$

$$w_{21} = 3/7; b_2 = -7.4$$

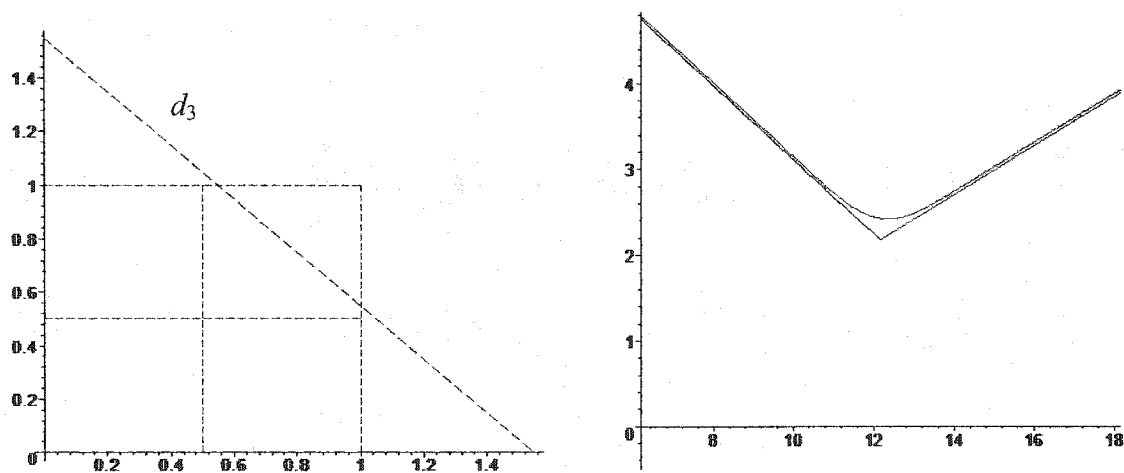
Notons que nous travaillons sur les paramètres du neurone de sortie et que nous fixons la valeur des paramètres des neurones cachés aux valeurs obtenues par les pentes des demi-droites formant la frontière bilinéaire.

Exemple 2.3:

$$w_{31} = 1; w_{32} = 1; b_3 = -1.55; a = 1$$

Figure 2.6 Frontière sans points d'intersection et $a = 1$ **Exemple 2.4:**

$$w_{31} = 1; w_{32} = 1, b_3 = -1.55 \text{ et } a = 5$$

Figure 2.7 Frontière sans points d'intersection et $a = 5$

Exemple 2.5:

$$w_{31} = 1.2; w_{32} = 1; b_3 = -1.6; a = 1$$

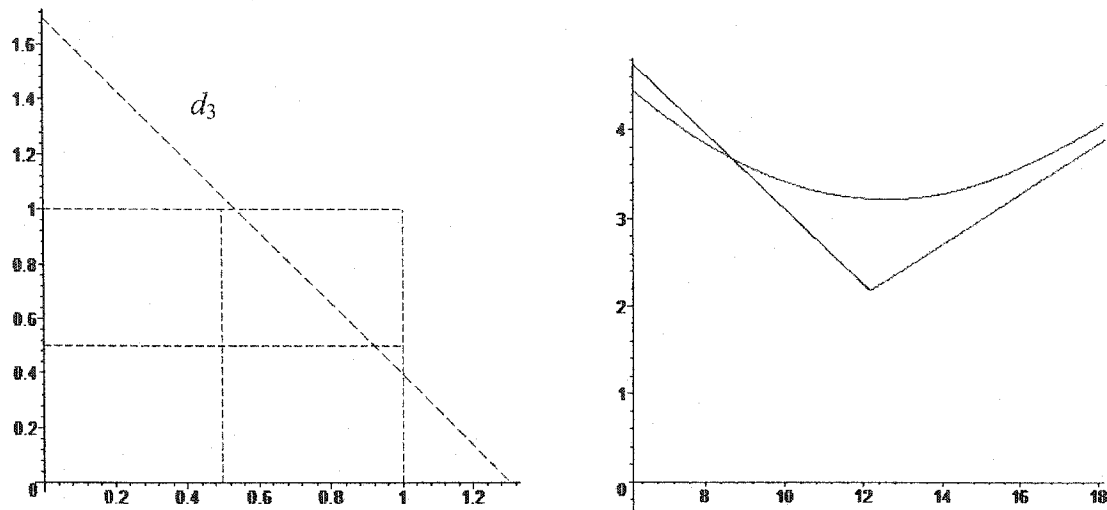


Figure 2.8 Frontière avec un seul point d'intersection

À l'aide des ces trois exemples, nous voyons que la forme de d_3 affecte directement celle de la courbe. Notons qu'entre l'exemple 2.3 et 2.4, seule la valeur du paramètre de la sigmoïde varie, a augmente de 1 à 5. La comparaison des deux figures (2.6, 2.7) permet de constater que plus la valeur de a augmente plus la frontière semble se rapprocher de la séparation désirée. C'est un résultat prévisible étant donné les remarques du premier chapitre.

Exemple 2.6:

Dans l'exemple 1.1 du chapitre précédent, nous avons trouvé que la valeur 3.3 est la valeur minimum de a qui sépare les données suivantes :

(5, 2.5), (6, 1.81), (7, 0.5), (5, 2.1), (7.2, 1.25), (7.5, 1.5), (8, 1.3), (7.4, 1.2), (9, 3.5),

(9, 4), (10, 2) et (9, 2).

Nous voulons tracer ces différentes données et à l'aide de l'équation 2.6 tracer la frontière afin de savoir si nous allons obtenir une bonne séparation.

Les équations des deux demi-droites formant la frontière sont :

$$y = -2/5 x + 4.2$$

$$y = 3/5 x - 3.2$$

Les coordonnées des points définissant d_3 sont : (0.9902, 0.3497) et (0.4182, 0.9995).

À l'aide de ces coordonnées nous pouvons déduire que :

La pente de d_3 est -1.1517.

La droite d_3 coupe l'axe des ordonnées en $y = 1.4812$.

Nous pouvons donc trouver les valeurs de w_{31} , w_{32} et b_3 que voici :

$$w_{31} = 1.5117, w_{32} = 1, \text{ et } b_3 = -1.4812$$

Voici la frontière obtenue ainsi que les différentes données :

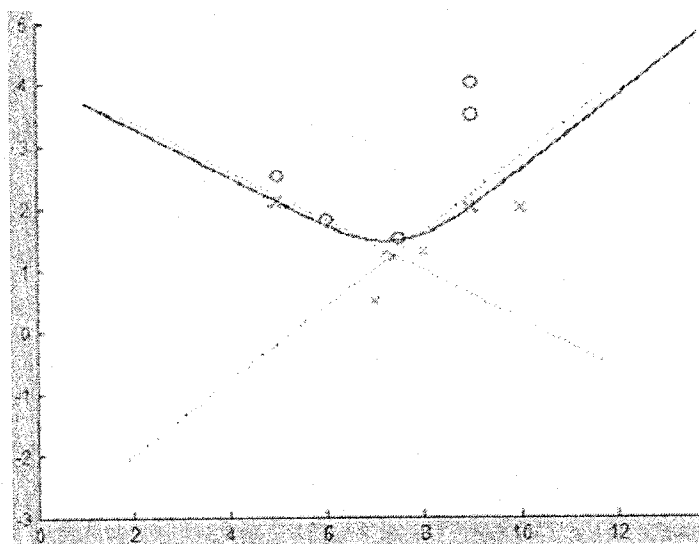


Figure 2.9 Séparation des données avec le a minimum

La figure 2.9 nous permet de voir que la courbe obtenue permet de séparer les deux classes. Tous les points de C_1 (cercles) sont au dessus de la courbe ($V_3 > 0$) tandis que le reste des points, ceux de C_2 , sont soit sur la courbe soit en dessous ($V_3 \leq 0$). La valeur a égale à 3.3 obtenue est donc suffisante pour obtenir la séparation désirée.

2.1.3 Frontière du réseau et asymptotes

Dans cette partie, nous voulons savoir s'il existe des valeurs pour w_{31} , w_{32} et b_3 qui vont permettre d'obtenir une frontière qui suit asymptotiquement les deux demi-droites.

Nous allons commencer par une étude dans l'espace caché.

Puisque nous voulons que la courbe suive asymptotiquement les deux droites, il faut que d_3 se rapproche des segments d'_1 et d'_2 sans les couper. L'espace caché qui semble le mieux répondre à ces exigences est le suivant :

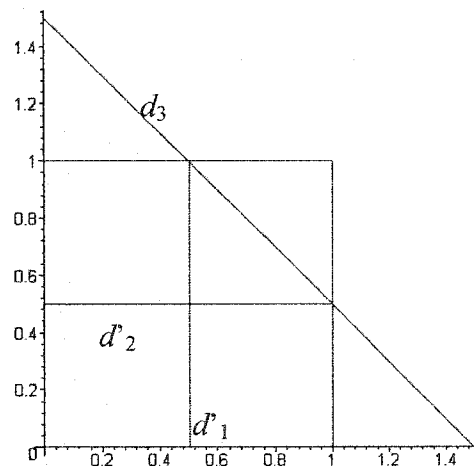


Figure 2.10 Asymptotes et espace caché

Pour obtenir cette espace caché nous devons avoir :

$$w_{31} = w_{32} = 1 \text{ et } b_3 = -1.5$$

Nous remarquons dans la figure 2.10 que nous sommes à la limite d'avoir un point d'intersection entre d_3 et chacun des deux segments d'_1 et d'_2 . Les systèmes 2.9 et 2.12 permettent de vérifier que pour ces valeurs il n'y a de points d'intersection ni entre d_1 et la courbe ni entre d_2 et la courbe, en effet :

$$\left(-\frac{w_{31} + 2w_{32} + 2b_3}{w_{31} + 2b_3} \right) = \left(-\frac{1+2-3}{1-3} \right) = 0$$

et

$$\left(-\frac{w_{32} + 2w_{31} + 2b_3}{w_{32} + 2b_3} \right) = \left(-\frac{1+2-3}{1-3} \right) = 0$$

Nous voulons maintenant démontrer que ces valeurs permettent à la courbe de se rapprocher asymptotiquement des demi-droites.

1^{er} Cas :

Supposons que les deux droites ont des pentes de signes opposés, nommons d_1 celle à pente positive.

Voici l'équation 2.1 :

$$V_3 = w_{31} \frac{1}{1 + e^{[-a(w_{11}x + y + b_1)]}} + w_{32} \frac{1}{1 + e^{[-a(w_{21}x + y + b_2)]}} + b_3$$

Soit un point (x_1, y_1) qui se situe très loin du point d'intersection des deux demi-

droites, mais sur la droite d_1 .

Voici les équations applicables aux coordonnées de ce point :

$$w_{11} x_1 + w_{12} y_1 + b_1 = 0$$

et

$$w_{21} x_1 + w_{22} y_1 + b_2 > 0$$

ou

$$w_{21} x_1 + w_{22} y_1 + b_2 < 0$$

Étant donné que la droite d_1 est celle à pente positive, et afin de se situer dans le demi-plan de d_2 qui nous intéresse (Figure 1.6), nous étudions le cas où :

$$w_{21} x_1 + w_{22} y_1 + b_2 > 0$$

Si nous travaillons pour des valeurs limites, donc si :

$$w_{21} x_1 + w_{22} y_1 + b_2 \rightarrow +\infty$$

Alors, pour le point (x_1, y_1) d'intérêt, l'équation 2.1 se réécrit de la forme suivante :

$$V_3 \rightarrow w_{31} \frac{1}{1+1} + w_{32} \frac{1}{1+0} + b_3$$

Si $w_{31} = w_{32} = 1$ et $b_3 = -1.5$, alors

$$V_3 \rightarrow 1 \frac{1}{1+1} + 1 \frac{1}{1+0} - \frac{3}{2}$$

donc

$$V_3 \rightarrow 0$$

Nous venons de montrer que V_3 tend vers 0 pour un point (x_1, y_1) sur la droite d_1 et loin

du point d'intersection des demi-droites. Donc, pour ces valeurs des paramètres de sortie, la frontière générée par le réseau suit et se rapproche de d_1 . Cependant, nous avons montré qu'il n'y a pas d'intersection entre cette droite et la courbe, nous pouvons donc déduire que loin du point d'intersection de d_1 avec d_2 la courbe se rapproche asymptotiquement de d_1 . De la même manière, nous pouvons démontrer que pour ces mêmes valeurs, la courbe suit asymptotiquement la demi-droite d_2 .

2^{ème} Cas :

Supposons que les deux droites ont des pentes de même signe (Figure 2.11).

Dans ce cas et sachant que nous travaillons dans la partie P^+ du plan, nous pouvons montrer que la courbe suivra asymptotiquement les deux demi-droites en utilisant le même raisonnement que dans le premier cas.

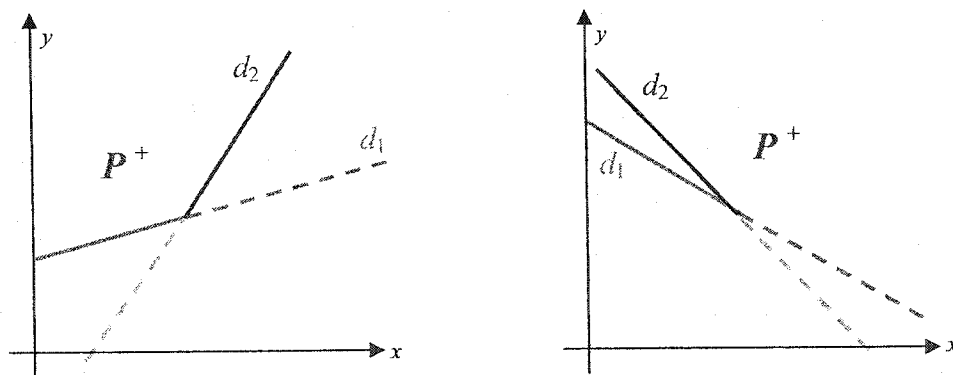


Figure 2.11 Droites de pentes de même signe

2.2 Généralisation du résultat

Nous avons obtenu une équation explicite de la forme $y = f(x)$ pour le réseau, à deux

neurones cachés. Cette équation nous a permis de tracer la frontière générée par le réseau et d'obtenir des informations concernant l'effet des différents paramètres sur cette frontière. Étant donné l'utilité de cette équation, nous voulons essayer de la généraliser. Nous voulons donc savoir si nous pouvons trouver une équation explicite même en rajoutant un troisième neurone dans la couche cachée. Le réseau obtenu est alors (figure 2.12) :

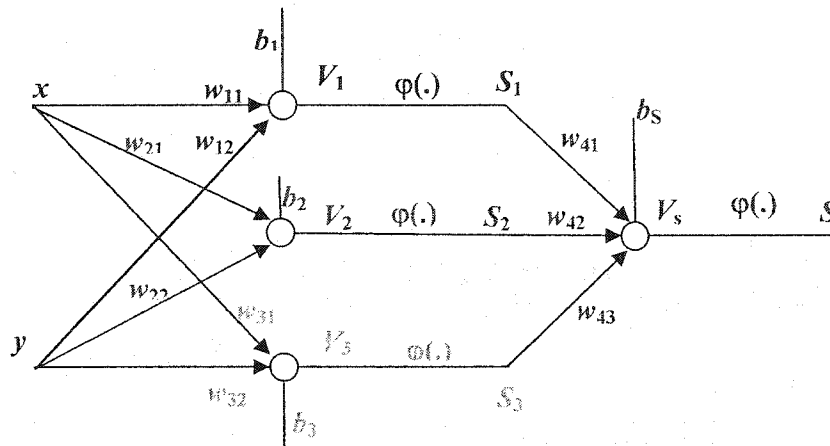


Figure 2.12 Réseau à trois neurones cachés

2.2.1 Lemme 3

Soit un réseau dont l'architecture est donnée par la figure 2.12 et dont la fonction d'activation est la sigmoïde. Nous pouvons trouver une équation explicite de la frontière engendrée par le réseau sachant que les poids issus de l'une des entrées sont unitaires.

Preuve :

De la même façon que nous avons procédé pour le lemme 2, nous pouvons donner les équations des neurones cachés :

$$V_1 = w_{11}x + w_{12}y + b_1$$

$$V_2 = w_{21}x + w_{22}y + b_2$$

$$V_3 = w_{31}x + w_{32}y + b_3$$

Et

$$S_1 = \varphi(V_1)$$

$$S_2 = \varphi(V_2)$$

$$S_3 = \varphi(V_3)$$

Comme nous l'avons fait dans le premier chapitre pour obtenir les équations des demi-droites, nous allons poser :

$$w_{12} = w_{22} = w_{32} = 1$$

Notons que cette restriction (les poids issus de l'une des entrées sont unitaires) est identique à celle que nous avons dans le lemme 2.

Posons que

$$w_{11} = W_1; w_{21} = W_2; w_{31} = W_3$$

$$b_1 = B_1; b_2 = B_2; b_3 = B_3$$

Nous pouvons écrire

$$V_1 = W_1x + y + B_1$$

$$V_2 = W_2x + y + B_2$$

$$V_3 = W_3x + y + B_3$$

Pour le neurone de la couche de sortie,

$$V_S = w_{41}S_1 + w_{42}S_2 + w_{43}S_3 + b_S$$

$$V_S = w_{41} \frac{1}{1 + e^{[-a(W_1x + y + B_1)]}} + w_{42} \frac{1}{1 + e^{[-a(W_2x + y + B_2)]}} + w_{43} \frac{1}{1 + e^{[-a(W_3x + y + B_3)]}} + b_s \quad 2.14$$

Soit

$$A'' = W_1x + B_1$$

$$B'' = W_2x + B_2$$

$$C'' = W_3x + B_3$$

Donc l'équation 2.14 devient:

$$w_{41} \frac{1}{1 + e^{-a(A'' + y)}} + w_{42} \frac{1}{1 + e^{-a(B'' + y)}} + w_{43} \frac{1}{1 + e^{-a(C'' + y)}} + b_s = 0$$

En posant,

$$D'' = e^{[-aA'']}$$

$$E'' = e^{[-aB'']}$$

$$F'' = e^{[-aC'']}$$

$$w_{41} \frac{1}{1 + D''e^{-ay}} + w_{42} \frac{1}{1 + E''e^{-ay}} + w_{43} \frac{1}{1 + F''e^{-ay}} + b_s = 0 \Rightarrow$$

$$\frac{w_{41}[1 + E''e^{-ay}][1 + F''e^{-ay}] + w_{42}[1 + D''e^{-ay}][1 + F''e^{-ay}] + w_{43}[1 + D''e^{-ay}][1 + E''e^{-ay}]}{[1 + D''e^{-ay}][1 + E''e^{-ay}][1 + F''e^{-ay}]} + b_s = 0$$

$$\Rightarrow w_{41}[1 + E''e^{-ay}][1 + F''e^{-ay}] + w_{42}[1 + D''e^{-ay}][1 + F''e^{-ay}] + w_{43}[1 + D''e^{-ay}][1 + E''e^{-ay}] + b_s[1 + F''e^{-ay}][1 + D''e^{-ay}][1 + E''e^{-ay}] = 0$$

Après développement, nous obtenons :

$$\begin{aligned}
 & w_{41} + w_{41}F''e^{-ay} + w_{41}E''e^{-ay} + w_{41}E''F''e^{-2ay} + w_{42} + w_{42}D''e^{-ay} + w_{42}F''e^{-ay} \\
 & + w_{42}D''F''e^{-2ay} + w_{43} + w_{43}D''e^{-ay} + w_{43}E''e^{-ay} + w_{43}D''E''e^{-2ay} + b_s + b_sF''e^{-ay} \\
 & + b_sE''e^{-ay} + b_sD''e^{-ay} + b_sE''F''e^{-2ay} + b_sE''F''e^{-2ay} + b_sD''F''e^{-2ay} \\
 & + b_sD''E''e^{-2ay} + b_sD''E''F''e^{-3ay} = 0
 \end{aligned} \tag{2.15}$$

Soient,

$$T_1'' = w_{41}F'' + w_{41}E'' + w_{42}D'' + w_{42}F'' + w_{43}D'' + w_{43}E'' + b_sF'' + b_sE'' + b_sD''$$

$$T_2'' = w_{41}F''E'' + w_{42}D''F'' + w_{43}D''E''$$

$$T_3'' = b_sD''E''F''$$

$$T_4'' = w_{41} + w_{42} + w_{43} + b_s$$

Nous pouvons donc réécrire l'équation 2.15 sous la forme suivante :

$$T_4'' + T_3''e^{-3ay} + T_2''e^{-2ay} + T_1''e^{-ay} = 0$$

Si

$$X = e^{-y}$$

$$\Rightarrow T_4'' + T_3''X^{3a} + T_2''X^{2a} + T_1''X^a = 0$$

Soit $Z = X^a$

$$\Rightarrow T_4'' + T_3''Z^3 + T_2''Z^2 + T_1''Z = 0$$

L'équation ci-dessus est du troisième degré ($T_3 \neq 0$). Elle admet donc trois solutions Z_1 , Z_2 et Z_3 , qui ne sont pas nécessairement toutes réelles.

En faisant des substitutions inverses, nous obtenons :

$$y_1 = -\ln \sqrt[a]{Z_1} = -\frac{1}{a} \ln(Z_1)$$

$$y_2 = -\ln \sqrt[a]{Z_2} = -\frac{1}{a} \ln(Z_2)$$

$$y_3 = -\ln \sqrt[a]{Z_3} = -\frac{1}{a} \ln(Z_3)$$

Il y a alors trois possibilités pour l'équation explicite $y = f(x)$. Parmi ces possibilités, il faut trouver celle (ou celles) qui permettra d'obtenir une équation explicite qui soit valide et réelle. Cette équation représentera la frontière générée par le réseau.

Nous venons de montrer que même en ajoutant un troisième neurone dans la couche cachée nous sommes capables de trouver une équation explicite pour la frontière du réseau. Cette équation explicite découle de la résolution d'une équation du troisième degré.

Nous venons de montrer qu'il est possible de trouver une fonction explicite de la frontière que va engendrer le réseau même si nous ajoutons un troisième neurone dans sa couche cachée.

2.2.2 Théorème 3

Soit un réseau dont l'architecture est donnée par la figure 2.12 et dont la fonction d'activation est la sigmoïde. Si les poids issus de l'une des entrées sont unitaires, nous pouvons trouver une équation explicite de la frontière engendrée par le réseau tant que le nombre de neurones cachés est strictement inférieur à 5.

Preuve :

Si nous avons quatre neurones cachés dans le réseau de la figure 2.12, alors en utilisant la même démarche que celle du lemme précédent, nous allons aboutir à une équation du quatrième degré. Sachant qu'il est possible de résoudre une équation du quatrième degré de façon analytique (théorème de Galois-Abel), nous pouvons alors trouver l'équation explicite recherchée. Finalement, si nous ajoutons encore un cinquième neurone, nous obtiendrons une équation du cinquième degré. Le fait que nous ne pouvons pas, en général, trouver les solutions d'une équation du cinquième degré implique que nous ne pouvons pas toujours trouver une équation de la frontière générée par ce réseau.

2.3 Justification du choix du repère

Nous avons obtenu une fonction explicite qui représente la frontière formée par le réseau. Cependant, et comme nous l'avons déjà remarqué, pour que notre étude analytique puisse ce faire, il est indispensable que la frontière désirée représente une fonction. En pratique, si nous avons une frontière bilinéaire à approximer il est possible qu'elle ne représente pas une fonction, voici deux exemples (figure 2.13 et 2.14) :

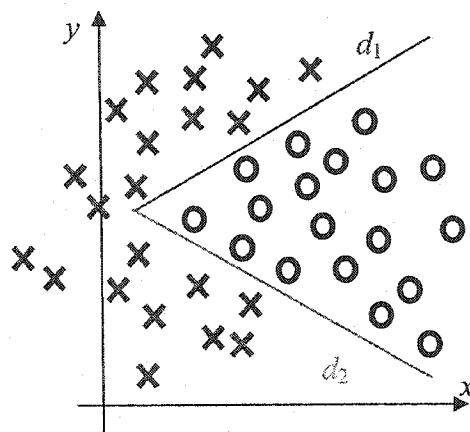


Figure 2.13 Frontière qui n'est pas une fonction: droites de pentes opposées

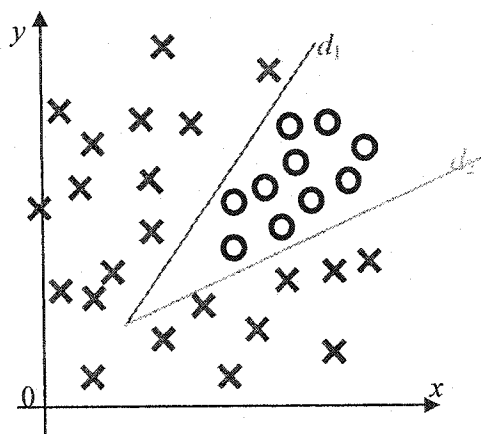


Figure 2.14 Frontière qui n'est pas une fonction : droites de pentes de même signe

Il est important de remarquer que pour faire l'étude théorique de la frontière, cette dernière doit former une fonction. Supposons que nous cherchons à approximer la frontière de la figure 2.14. Cette frontière n'étant pas une fonction, nous devons d'abord faire le changement de repère afin de satisfaire à cette exigence. Une fois ce changement effectué nous pourrions utiliser l'équation 2.6 et les différents résultats qui en découlent.

De plus, comme nous le verrons dans le prochain chapitre, il nous sera possible d'optimiser les paramètres du réseau et d'obtenir la plus petite valeur théorique de l'erreur, cette valeur pourra nous servir, par exemple, de référence. En pratique, lors de l'apprentissage du réseau, nous n'avons pas besoin que la frontière bilinéaire soit une fonction vu que l'apprentissage du réseau ne dépend pas de cette condition. Cependant, une fois l'apprentissage terminé nous pourrions alors calculer l'erreur pratique obtenue par ce réseau et comparer cette dernière à la valeur théorique. Cette comparaison nous permet, par exemple, d'évaluer les performances de l'algorithme d'apprentissage utilisé. Afin de pouvoir procéder à notre étude théorique, nous devons être capable de faire les transformations nécessaires à notre repère pour qu'il satisfasse notre condition (frontière formant une fonction). Nous allons commencer par présenter des formules qui permettent de faire un changement de repère, par la suite, nous expliciterons un moyen qui va permettre de savoir dans quel cas il faut faire un changement de repère et comment le faire afin de se ramener au cas désiré.

Formules de rotations d'axes :

Soit un repère xy d'origine O ; si nous faisons subir aux axes x et y une rotation autour de l'origine O d'un angle Φ , nous obtenons un nouveau repère $x'y'$. Nous pouvons obtenir les coordonnées (x', y') d'un point P dans le plan $x'y'$ à partir de ses coordonnées (x, y) dans le repère initial. Les équations sont :

$$x = x' \cos(\Phi) - y' \sin(\Phi)$$

$$y = x' \sin(\Phi) + y' \cos(\Phi)$$

Et

$$x' = x \cos(\Phi) + y \sin(\Phi)$$

$$y' = -x \sin(\Phi) + y \cos(\Phi)$$

Formules de translation d'axes :

Si (x, y) sont les coordonnées d'un point P dans le plan xy et (x', y') les coordonnées de P dans le plan $x'y'$ dont l'origine se trouve en (h, k) alors nous pouvons écrire les équation suivantes :

$$x = x' + h$$

$$y = y' + k$$

Et

$$x' = x - h$$

$$y' = y - k$$

À l'aide des formules que nous venons de présenter, nous pouvons trouver les équations des droites dans un nouveau repère.

2.3.1 Changement de repère

Nous voulons trouver un critère mathématique qui va permettre de savoir s'il est nécessaire de faire un changement de repère.

Soit un plan xy et une droite d'équation

$$y = a_1x + c_1$$

Si $a_1 > 0$, donc si la pente est positive, alors le demi-plan positif (P^+) est à la « gauche » de la droite (Figure 2.15).

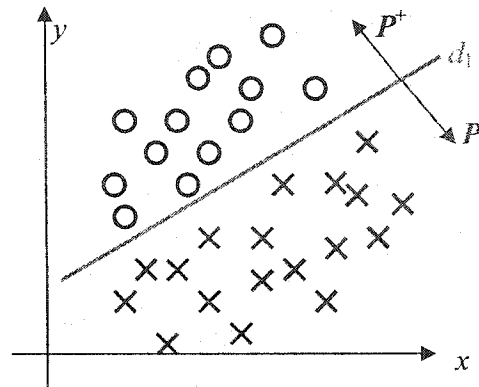


Figure 2.15 Droite à pente positive et signe des demi-plans

Si $a_1 < 0$, donc si la pente est négative alors le demi-plan positif est à la « droite » de la droite (Figure 2.16).

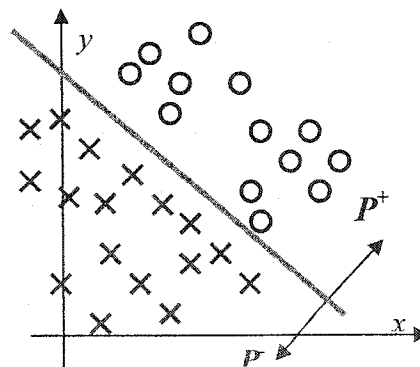


Figure 2.16 Droite à pente négative et signe des demi-plans

Note :

- Si $a_1 = 0$, alors le demi-plan positif est « au dessus » de la droite.

- Si $a_1 = \infty$, alors le demi-plan positif est à la « droite » de la droite.

Les deux figures ci-dessous permettent de voir des séparations de deux classes dont l'une appartient à l'intersection des demi-plans positifs.

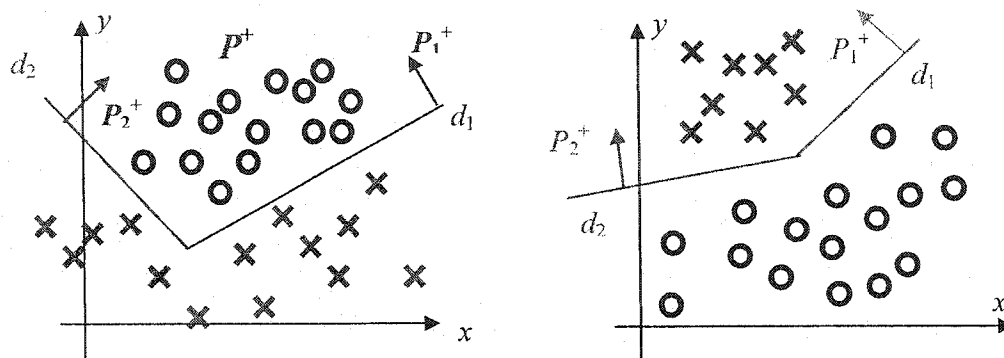


Figure 2.17 Union de deux demi-plans positifs

Après avoir analysé les figures 2.15, 2.16 et 2.17, nous pouvons affirmer que pour que la séparation de nos deux classes forme une fonction, il faut que l'une des classes appartienne à l'intersection des deux demi-plans positifs ou des deux demi-plans négatifs tandis que l'autre classe doit appartenir au reste du plan. Si cette condition n'est pas satisfaite, alors la frontière bilinéaire à approximer n'est pas une fonction. Dans le reste de notre étude, nous ne prendrons que le cas où nous voulons isoler l'intersection des demi-plans positifs du reste du plan. Cependant, si nous voulons isoler l'union des deux demi-plans négatifs, il nous suffit d'une simple rotation afin de se retrouver dans le cas étudié ci-dessous.

Nous allons maintenant traduire nos résultats en équations mathématiques :

Soit nos deux droites d'équations :

$$y = -a_1x - c_1$$

$$y = -a_2x - c_2$$

et deux points de coordonnées (Px_1, Py_1) et (Px_2, Py_2) appartenant respectivement aux classes C_1 et C_2 .

Soient les systèmes suivants :

$$\left\{ \begin{array}{l} M_1 = a_1 Px_1 + Py_1 + c_1 \\ M_2 = a_2 Px_1 + Py_1 + c_2 \\ M_3 = a_1 Px_2 + Py_2 + c_1 \\ M_4 = a_2 Px_2 + Py_2 + c_2 \end{array} \right. \quad 2.16$$

$$\left\{ \begin{array}{l} (M_1, M_2) > 0 \\ \text{et} \\ (M_3, M_4) < 0 \end{array} \right. \quad 2.17$$

$$\left\{ \begin{array}{l} (M_1, M_2) < 0 \\ \text{et} \\ (M_3, M_4) > 0 \end{array} \right. \quad 2.18$$

Les deux systèmes 2.17 et 2.18 permettent de s'assurer que l'une des deux classes appartient à l'intersection des deux demi-plans positifs, tandis que l'autre classe

appartient au reste du plan, donc que la frontière bilinéaire forme une fonction dans le plan.

Dans le cas où les deux systèmes ne sont pas satisfaits, alors des rotations et des translations du repère seront nécessaires. Nous allons maintenant expliquer comment faire ces différentes transformations.

D'abord si le point d'intersection des deux droites n'est pas confondu avec l'origine du repère, nous faisons une translation qui nous permet de positionner l'origine du nouveau repère à l'intersection des deux droites (Figure 2.18).

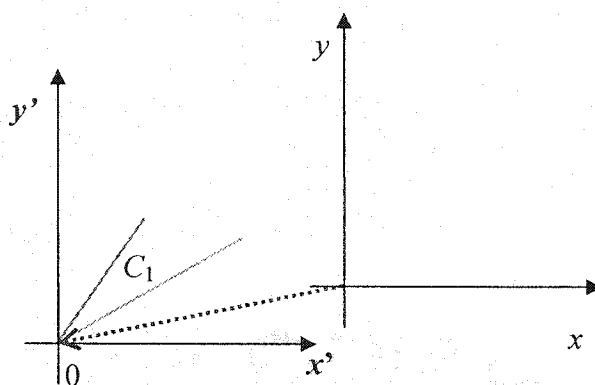


Figure 2.18 Translation d'un repère

Une fois cette translation faite nous travaillons alors dans un nouveau repère $x'y'$ dont l'origine est confondue avec le point d'intersection des deux droites. Voici les différents cas possibles dans lesquels nous pouvons nous retrouver :

1^{er} cas

Si les deux droites ont des pentes positives alors nous devons calculer la valeur de l'angle β entre la droite de plus grande pente et l'axe des ordonnées. Ensuite, calculer la valeur de l'angle γ formé par les deux droites, puis faire une rotation au repère d'un angle α , tel que $\alpha \in]\beta ; \beta + \gamma[$.

Après cette rotation l'une des droites se situera dans le demi-plan $x < 0$, et l'autre sera dans le demi-plan $x > 0$. Voici la figure 2.19 pour comprendre la démarche expliquée.

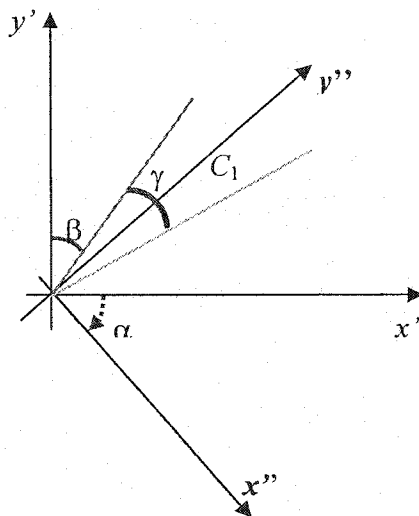


Figure 2.19 Rotation avec deux droites à pentes positives

2^{ième} cas

Si les deux droites sont de pentes négatives (figure 2.20) alors nous faisons subir au repère une rotation de $-\pi/2$. Ceci nous permettra de nous ramener au 1^{er} cas ci-dessus.

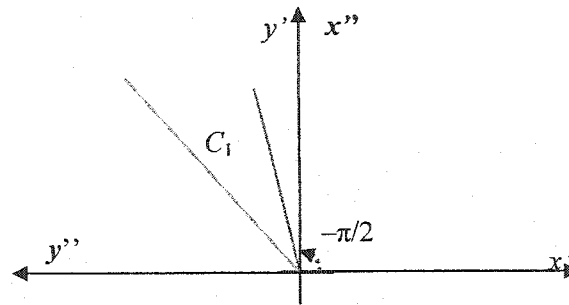


Figure 2.20 Rotation avec deux droites à pentes négatives

3^{ème} cas

Dans le cas où nous avons deux droites de pente de signes opposés, il nous suffit de faire une rotation de $-\pi/2$ si la classe à isoler appartient au demi-plan $x' < 0$ (Figure 2.21), et de $\pi/2$ si la classe à isoler appartient au demi-plan $x' > 0$ (Figure 2.22). Une fois ces rotations faites, nous nous retrouvons avec un repère $x'y'$ dans lequel les deux demi-droites forment une fonction.

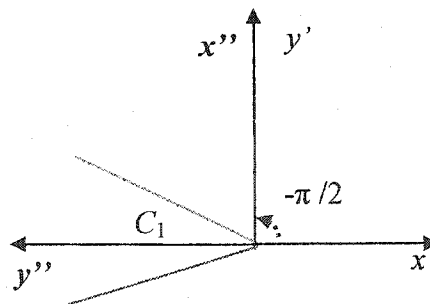
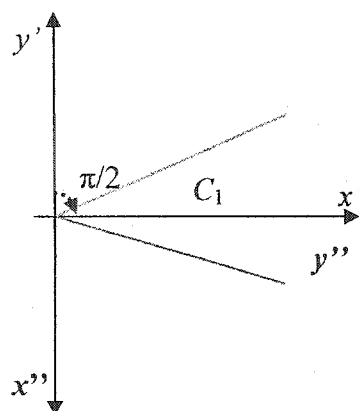


Figure 2.21 Rotation de $-\pi/2$

Figure 2.22 Rotation de $\pi/2$ 4^{ème} cas

Si nous avons deux droites dont les pentes sont de signes opposés (figure 2.23) et que nous voulons séparer l'intersection des deux demi-plans négatifs, nous devons d'abord faire une rotation de π ou $-\pi$.

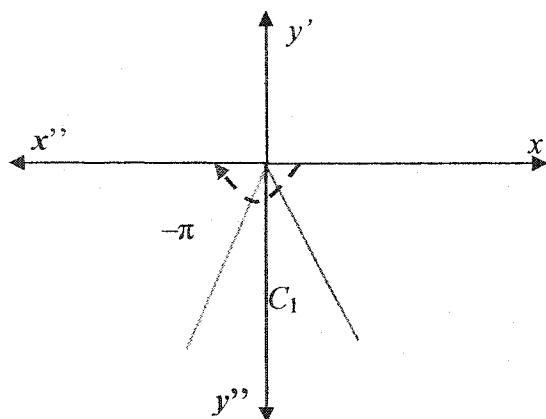


Figure 2.23 Séparation des deux demi-plans négatifs

Ces différents cas couvrent les différentes possibilités. À l'aide des transformations que nous avons énumérées, nous sommes capables de faire tous les changements de repère nécessaires pour obtenir une séparation qui est une fonction, une fois ces changements faits, nous pourrons alors utiliser l'équation 2.6 pour faire l'étude théorique de la frontière.

CHAPITRE 3 OPTIMISATION DES PARAMÈTRES

Dans le chapitre précédent nous nous sommes attardés sur la recherche de l'équation 2.6. Cette équation a permis d'obtenir plusieurs résultats intéressants comme les résultats concernant les asymptotes et les équations des points d'intersection. À présent, nous cherchons à obtenir le réseau optimal, c'est-à-dire celui qui, pour une valeur fixe de a , reproduit avec le plus petit pourcentage d'erreur une frontière bilinéaire. Définissons d'abord ce pourcentage d'erreur.

Soit le plan xy , soit aussi d_1 et d_2 qui forment la frontière bilinéaire désirée et soit $f(x)$ la courbe obtenue par le réseau. L'espace de variation est donné par $x \in [a, b]$ et $y \in [c, d]$. Le pourcentage d'erreur est défini par le rapport entre la surface mal classifiée (surface pleine dans la figure 3.1) et la surface totale de variation (rectangle).

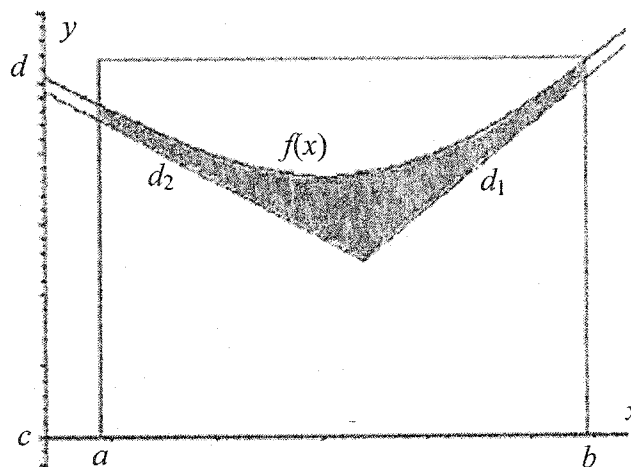


Figure 3.1 Surface d'erreur et espace de définition

3.1 Pourcentage d'erreur dans l'espace caché

Nous cherchons maintenant à calculer le pourcentage d'erreur dans l'espace caché.

Il est facile de voir, par exemple, dans la figure 3.2 que la surface d'erreur est représentée par le triangle hachuré.

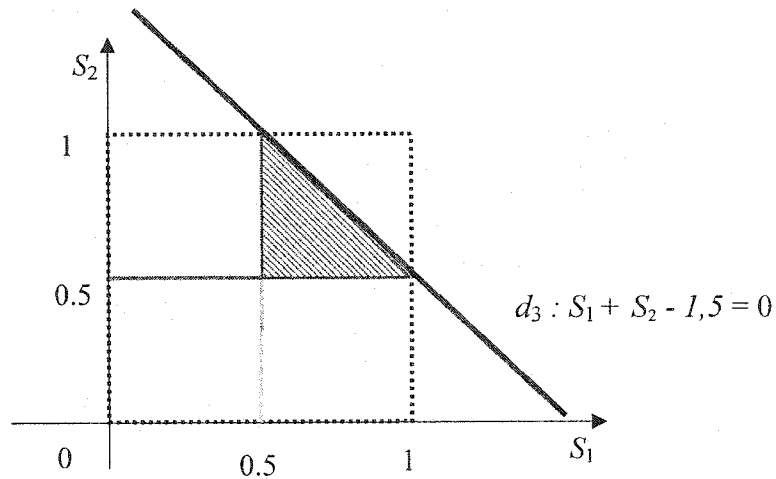


Figure 3.2 Espace caché et surface d'erreur

Si nous calculons le pourcentage d'erreur en prenant le rapport de la surface d'erreur sur la surface totale nous obtenons une valeur de 12.5% ($1/8$ du carré). Cependant, après une brève réflexion nous nous rendons compte que dans cet espace nous ne pouvons pas calculer l'erreur de cette façon. En effet, nous devons d'abord calculer la fonction de densité des points dans $S_1 S_2$; ensuite, nous devons chercher l'espace image de l'espace de variation. Afin de clarifier ces deux notions nous allons les développer dans les sections suivantes. Nous commencerons par l'étude de la fonction de densité, puis nous suivrons par celle de l'espace image.

3.1.1 Étude de la fonction de densité dans l'espace caché.

Soit un plan xy et R_1 une région de ce plan.

Nous pouvons dire que la densité des points, ou que la fonction de densité $f_{x,y}(x,y)$, est uniforme dans la région R_1 du plan si la probabilité d'avoir une donnée dans n'importe quel point de cette région est toujours égale, donc constante.

Nous avons écrit un programme MatLab qui nous permet de visualiser une région de l'espace des entrées ainsi que son image dans l'espace caché. La densité est considérée uniforme dans l'espace des entrées.

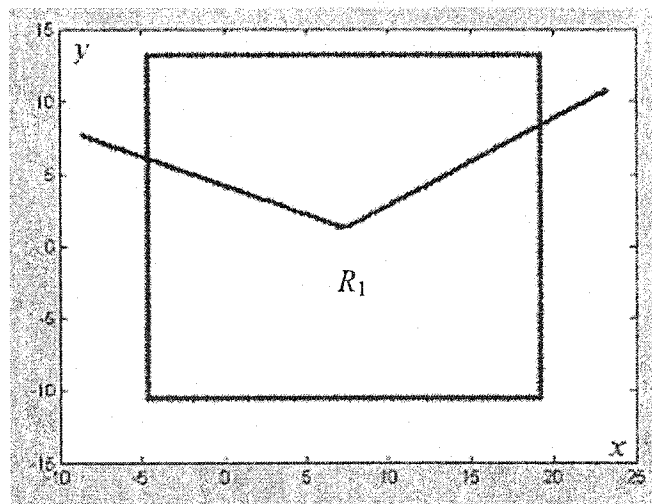


Figure 3.3 Espace de variation R_1

Dans l'espace de variation R_1 (Figure 3.3), nous prenons uniformément 14 400 points (120×120) et nous traçons leurs images dans l'espace caché pour deux valeurs du paramètre a de la sigmoïde (figure 3.4 et 3.5).

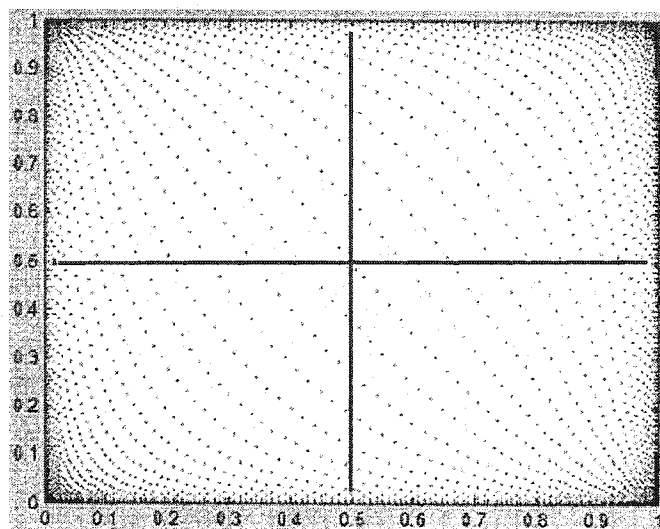


Figure 3.4 Espace image avec $a = 1$

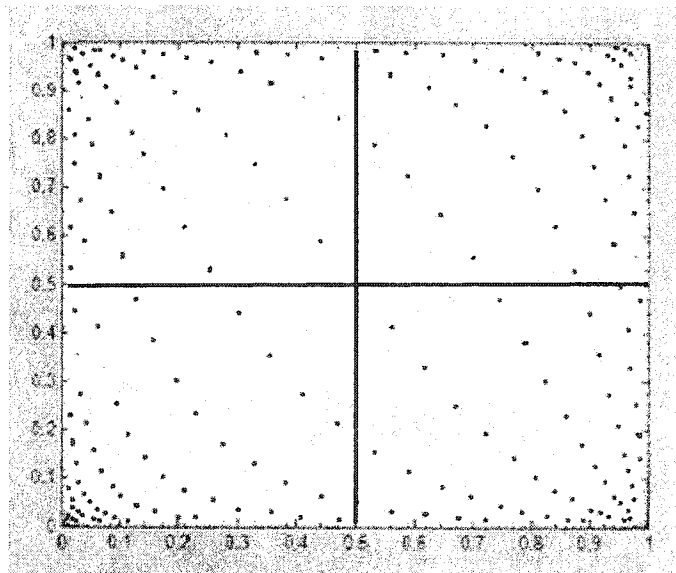


Figure 3.5 Espace image avec $a = 3$

Nous remarquons, d'abord, que la densité des points images n'est pas uniforme dans le carré. Elle est plus élevée dans les coins du carré que dans son centre. Nous remarquons aussi qu'une variation du paramètre a affecte la dispersion des points images et la fonction de densité dans l'espace caché. Nous aurions pu prévoir ces résultats étant donné les analyses faites dans le premier chapitre. À l'aide de la remarque qui suit nous allons être capable de calculer la fonction de densité dans l'espace caché.

Remarque :

Soient X, Y deux variables aléatoires continues, et soit le système 3.1 suivant :

$$\begin{cases} V = g_1(X, Y) \\ W = g_2(X, Y) \end{cases} \quad 3.1$$

Supposons que

1) le système 3.1 possède comme solution unique le suivant :

$$\begin{cases} x = h_1(v, w) \\ y = h_2(v, w) \end{cases}$$

2) les fonctions g_1, g_2 possèdent des dérivées partielles continues $\forall (x, y)$ et que le jacobien de la transformation :

$$J(x, y) = \frac{\partial g_1}{\partial x} \frac{\partial g_2}{\partial y} - \frac{\partial g_2}{\partial x} \frac{\partial g_1}{\partial y}$$

est différent de zéro $\forall (x, y)$.

Alors nous pouvons écrire que

$$F_{V,W}(v,w) = F_{X,Y}(x,y) |J(x,y)|^{-1} \quad 3.2$$

L'équation 3.2 permet alors de calculer la fonction de densité après avoir fait un changement de variables. Ainsi, cette équation permet de calculer la fonction de densité dans l'espace caché. Ayant cette fonction il ne nous reste plus qu'à obtenir l'espace image de l'espace de variation.

3.1.2 Étude de l'espace image

Voici un exemple (figure 3.6) avec l'espace de variation qui est défini par les intervalles suivants :

$$x \in [6.4, 8.4]$$

$$y \in [0.25, 2.25]$$

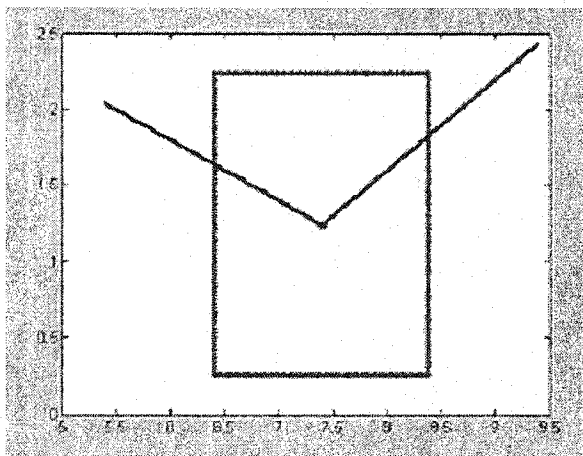


Figure 3.6 Espace $x \in [6.4, 8.4]$ et $y \in [0.25, 2.25]$

Dans la figure 3.7 nous traçons l'image de cet espace de variation pour $a = 1$.

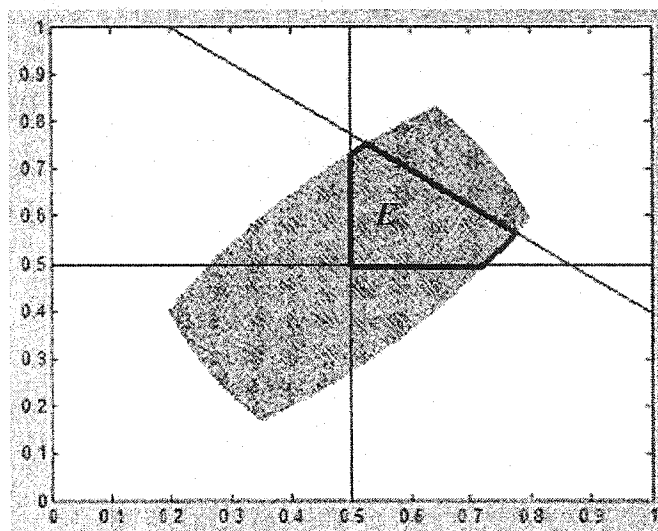


Figure 3.7 Délimitation de l'erreur (E) dans l'espace image pour $\alpha=1$

L'image est représentée par la forme géométrique hachurée dans la figure 3.7. Elle ne représente qu'une partie du carré unitaire. Pour formuler une expression de l'erreur, nous devons intégrer la fonction de densité sur la surface E . Cependant, les bornes d'intégrations et les limites de E étant dépendantes des paramètres du réseau, il est difficile de définir une expression générale et manipulable de l'intégrale à optimiser. Nous avons donc décidé de travailler dans l'espace xy .

3.2 Pourcentage d'erreur dans le plan des entrées

Dans le début de ce chapitre nous avons vu ce que représente le pourcentage d'erreur dans le plan des entrées (Figure 3.1). Nous allons maintenant définir une expression mathématique de la surface d'erreur.

Dans la figure 3.8 nous voyons la frontière à approximer, la courbe générée par le réseau, l'espace de variation et la surface d'erreur représentée par les surfaces pleines.

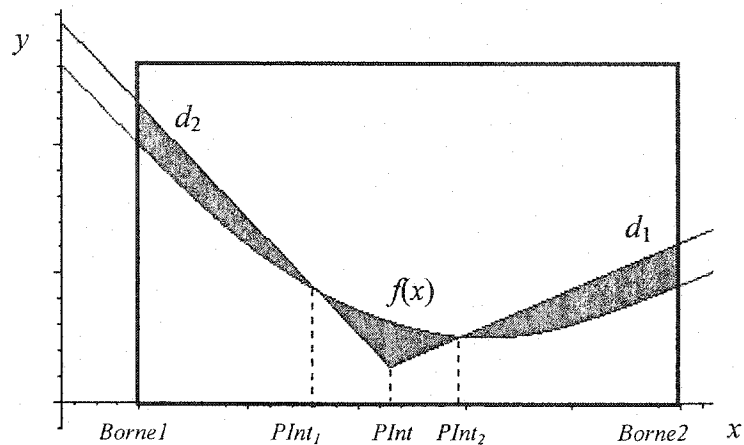


Figure 3.8 Bornes définissant la surface d'erreur

Voici l'expression mathématique de la surface d'erreur

$$\begin{aligned} \text{Surface Erreur} = & \int_{\text{Borne1}}^{\text{PInt}_1} (d_2(x) - f(x)) dx + \int_{\text{PInt}_1}^{\text{PInt}} (f(x) - d_2(x)) dx + \int_{\text{PInt}}^{\text{PInt}_2} (f(x) - d_1(x)) dx \\ & + \int_{\text{PInt}_2}^{\text{Borne2}} (d_1(x) - f(x)) dx \end{aligned} \quad 3.3$$

Où

- $d_1(x)$, $d_2(x)$ représentent les équations des deux demi-droites.
- $f(x)$ est la fonction qui représente la frontière (Équation 2.6).
- *Borne 1*, *Borne 2* représentent respectivement la borne inférieure et supérieure de l'espace de variation des données.

- $PInt_1$ est l'abscisse du point d'intersection entre la courbe et la demi-droite d_2 .
- $PInt_2$ est l'abscisse du point d'intersection entre la courbe et la demi-droite d_1 .
- $PInt$ est l'abscisse du point d'intersection entre d_1 et d_2 .

Si l'une des droites a une pente infinie, alors pour obtenir une expression de la surface d'erreur, il suffit d'une simple rotation du repère pour éviter d'avoir un problème avec les bornes de l'intégrale.

Notons que si la courbe ne coupe pas les deux droites ou ne coupe qu'une seule des deux droites alors, l'expression de la surface d'erreur n'est plus définie par l'équation 3.3. Cette dernière constatation entraîne une difficulté de plus lors de l'optimisation. Cependant nous pouvons simplifier l'équation 3.3 et l'écrire de la façon suivante :

$$\text{Surface Erreur} = \int_{\text{Borne1}}^{PInt} \text{Abs}[(d_2(x) - f(x))]dx + \int_{PInt}^{\text{Borne2}} \text{Abs}[(f(x) - d_1(x))]dx$$

où

$$\text{Abs}[f(x)-d_i(x)] = \text{Valeur Absolue}[f(x)-d_i(x)]$$

Nous voyons comment en introduisant la notion de valeur absolue il est possible de ne plus tenir compte des deux points d'intersection dans l'expression de l'erreur.

Notre but est de chercher à optimiser les performances du réseau. Pour ce faire, il faut chercher à minimiser la surface d'erreur en fonction des différents paramètres. En fait, dans l'expression 3.4 nous retrouvons les paramètres suivants : w_{31} , w_{32} , b_3 , w_{11} , b_1 , w_{21} , et b_2 , seulement les poids issues de l'entrée y (w_{12} et w_{22}) ont été fixées à un. Par suite si

nous minimisons la surface d'erreur à l'aide de l'expression 3.4 nous serons en mesure de trouver la valeur optimale de tous les paramètres suivants : w_{31} , w_{32} , b_3 , w_{11} , b_1 , w_{21} , et b_2 .

L'équation de $f(x)$ contenue dans les expressions 3.3 et 3.4 est assez complexe et il est impossible de trouver une valeur analytique de son intégrale. Nous devons donc procéder à quelques transformations en espérant obtenir les résultats escomptés.

3.3 Fonction explicite symétrique

Voici quelques opérations qui vont permettre de transformer l'équation 2.6 en une autre équation plus simple.

Lors de l'obtention de l'équation 2.6 nous avons utilisé des demi-droites dont les équations étaient de la forme :

$$y = px + c.$$

Cependant, avec les transformations adéquates, il est possible de transformer les deux demi-droites quelconques en deux demi-droites symétriques (Figure 3.9) par rapport aux ordonnées (dans le demi-plan $y > 0$), elles seront alors de la forme :

$$d_1 : y = wx$$

$$d_2 : y = -wx$$

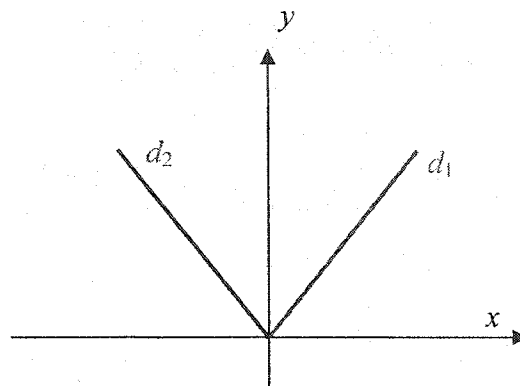


Figure 3.9 Droites symétriques par rapport aux ordonnées.

Avec ces opérations, l'équation 2.6 se simplifie, puisque nous posons :

$$b_1 = b_2 = 0$$

$$w_{11} = -w \text{ et } w_{21} = w$$

L'équation simplifiée se réécrit alors sous la forme suivante :

$$y = f(x) = -\frac{1}{a} \ln \left(\frac{-\frac{[(w_{31} + b_3)e^{[-a(-wx)]} + (w_{32} + b_3)e^{[-a(wx)]}]}{2b_3}}{\frac{\sqrt{[(w_{31} + b_3)e^{[-a(-wx)]} + (w_{32} + b_3)e^{[-a(wx)]}]^2 - 4b_3((b_3 + w_{32} + w_{31}))}}{2b_3}} \right) \quad 3.5$$

À titre d'exemple voici une courbe obtenue à l'aide de MAPLE, et avec les valeurs suivantes :

$$w = -5/7$$

$$w_{31} = 1.2, w_{32} = 1, b_3 = -1.4, a = 1$$

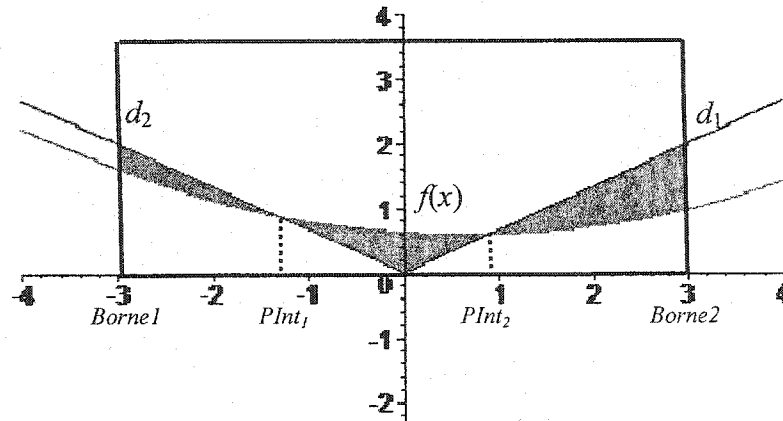


Figure 3.10 Surface d'erreur pour frontière quelconque

Nous avons choisi un cas général avec deux points d'intersection dont nous pouvons trouver les coordonnées à l'aide des formules 2.10 et 2.13.

3.3.1 Optimisation de la surface d'erreur

Dans le cas de la figure 3.10 l'expression de la surface d'erreur est :

$$\begin{aligned} \text{Surface Erreur} = & \int_{\text{Borne1}}^{\text{PInt}_1} (d_2(x) - f(x)) dx + \int_{\text{PInt}_1}^0 (f(x) - d_2(x)) dx + \int_0^{\text{PInt}_2} (f(x) - d_1(x)) dx \\ & + \int_{\text{PInt}_2}^{\text{Borne2}} (d_1(x) - f(x)) dx \end{aligned} \quad 3.6$$

Où

- $d_1(x)$, $d_2(x)$ représentent les équations des deux demi-droites symétriques.
- $f(x)$ est la fonction qui représente la frontière (Équation 3.5).
- *Borne 1*, *Borne 2* représentent respectivement la borne inférieure et supérieure de l'espace de variation des données.
- PInt_1 est l'abscisse du point d'intersection entre la courbe et la demi-droite d_2

dans le demi plan $x < 0$.

- $PInt_2$ est l'abscisse du point d'intersection entre la courbe et la demi-droite d_1 .

Comme nous l'avons fait dans la section précédente, nous pouvons simplifier l'équation 3.6 et obtenir :

$$\text{Surface Erreur} = E = \int_{\text{Borne 1}}^0 \text{Abs}[f(x) - d_2(x)]dx + \int_0^{\text{Borne 2}} \text{Abs}[f(x) - d_1(x)]dx \quad 3.7$$

L'expression 3.7 contient les quatre paramètres que voici :

$$w, w_{31}, w_{32} \text{ et } b_3$$

Malgré le fait que l'équation 3.5 soit moins complexe que l'équation 2.6, il est toujours impossible de trouver une équation analytique de son intégrale, rendant l'optimisation de la surface d'erreur difficile.

Nous allons tenter de trouver le minimum de l'expression 3.7 en fonction des paramètres du neurone de sortie : w_{31} , w_{32} et b_3 . En effet, nous fixons les poids des neurones cachés aux valeurs obtenues par les pentes des demi-droites formant la frontière bilinéaire, et cherchons à optimiser les valeurs du neurone de sortie.

Soient les dérivées partielles suivantes :

$$d'_1 = \frac{\delta E}{\delta w_{31}}; d'_2 = \frac{\delta E}{\delta w_{32}}; d'_3 = \frac{\delta E}{\delta b_3}; \quad 3.8$$

Afin de minimiser l'erreur il faut trouver les valeurs de w_{31} , w_{32} et b_3 solutions du système suivant :

$$\left\{ \begin{array}{l} d'_1 = \frac{\delta E}{\delta w_{31}} = 0 \\ d'_2 = \frac{\delta E}{\delta w_{32}} = 0 \\ d'_3 = \frac{\delta E}{\delta b_3} = 0 \end{array} \right. \quad 3.9$$

Les solutions de ce système représentent un minimum du pourcentage d'erreur. Cependant, il faudrait s'assurer de trouver le minimum global de E , une solution du système 3.9 pourrait ne représenter qu'un minimum local.

Exemple 3.1:

Soit deux demi-droites d'équations :

$$d_1 : y = 4/5 x$$

$$d_2 : y = -4/5 x$$

Le domaine de variation est :

$$x \in [-5 \ 5]$$

$$y \in [0 \ 8]$$

Le but est de trouver les paramètres optimaux du neurone de sortie.

Les équations des droites permettent de fixer les paramètres des deux neurones cachés :

$$w_1 = 4/5, b_1 = 0$$

$$w_2 = -4/5, b_2 = 0$$

Voici les deux demi-droites et le domaine de définition (figure 3.11).

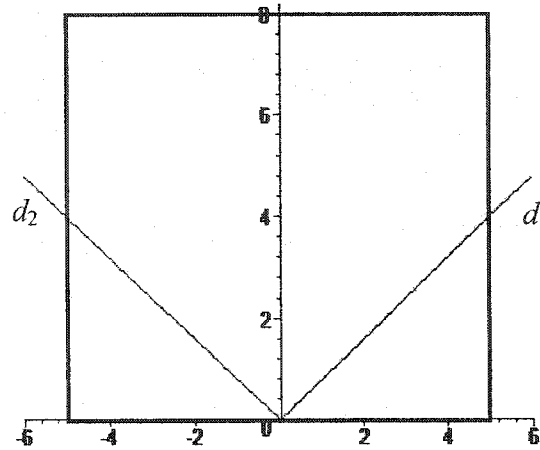


Figure 3.11 Frontière bilinéaire symétrique et domaine de variation

Vu que l'équation 3.5 est complexe, nous allons la manipuler à l'aide de MAPLE. Nous définissons alors l'expression de l'erreur (Équation 3.7) et les trois termes du système 3.9.

Le logiciel utilisé n'est pas capable de trouver une solution directe à ce système d'équations (Système 3.9). Nous devons trouver une autre façon de le résoudre.

Sachant que le domaine de variation et que la frontière à obtenir sont symétriques par rapport à l'axe des ordonnées, nous allons étudier le cas où la courbe est elle aussi symétrique par rapport à cet axe. Nous allons donc étudier le cas où $w_{31} = w_{32} = 1$, en effet avec ces valeurs nous obtenons (Équation 3.5) :

$$f(x) = f(-x)$$

Pour ces valeurs de w_{31} et w_{32} , voici la courbe qui représente l'erreur pour

$$b_3 \in [-1, -1.8]$$

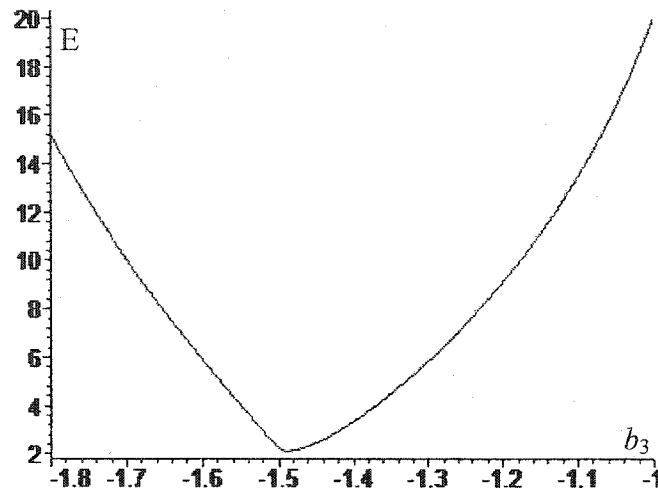


Figure 3.12 Courbe d'erreur.

C'est à l'aide de l'espace caché et sachant que $w_{31} = w_{32} = 1$, donc que la pente de d_3 est égale à 1, que nous avons choisi les bornes de b_3 .

La courbe de la figure 3.12 permet de voir que l'erreur passe par un minimum aux alentours de :

$$b_3 = -1.49$$

De plus, voici les représentations de d'_1 , d'_2 et d'_3 pour $b_3 \in [-1.45, -1.51]$:

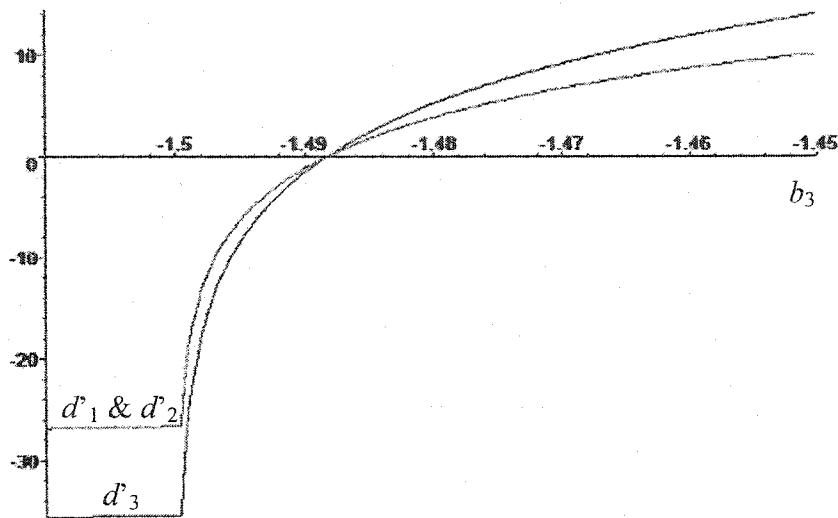


Figure 3.13 Valeurs de d'_1 , d'_2 et d'_3 en fonction de b_3 .

Nous ne voyons, dans cette figure, que deux courbes. Ceci s'explique par le fait que si $w_{31} = w_{32}$ alors $d'_1 = d'_2$. Nous remarquons cependant que ces courbes se coupent en un point et semblent être concourantes avec l'axe des abscisses aux alentours de $b_3 = -1.49$. Sachant que l'erreur atteint un minimum dans cet intervalle, nous pouvons alors affirmer que les trois termes d'_1 , d'_2 et d'_3 s'annulent en un même point dans cet intervalle.

Pour trouver une valeur plus précise pour b_3 nous pouvons utiliser MAPLE. Après plusieurs itérations, nous trouvons que :

- Si $b_3 = -1.4883$ nous obtenons

$$d'_1 = d'_2 = -0.0124552924$$

$$d'_3 = -0.01673772762$$

- Si $b_3 = -1.4882$ nous obtenons

$$d'_1 = d'_2 = 0.0498432323$$

$$d'_3 = 0.06698459112$$

Nous remarquons qu'à cause des approximations faites par MAPLE lors du calcul de l'intégrale, les valeurs des grandeurs d'_1 , d'_2 et d'_3 ne sont pas tout à fait exactes. Cependant étant donné l'analyse faite à l'aide des figures 3.12 et 3.13 nous pouvons affirmer que pour $b_3 \cong -1.4882$, alors d'_1 , d'_2 et d'_3 s'annulent simultanément et que le vecteur ($w_{31} = 1$, $w_{32} = 1$, $b_3 = -1.4882$) est solution du système 3.9.

Nous obtenons comme valeurs optimales :

$$w_{31} = w_{32} = 1, b_3 = -1.4882$$

Voici la courbe obtenue avec les paramètres optimaux (figure 3.14) :

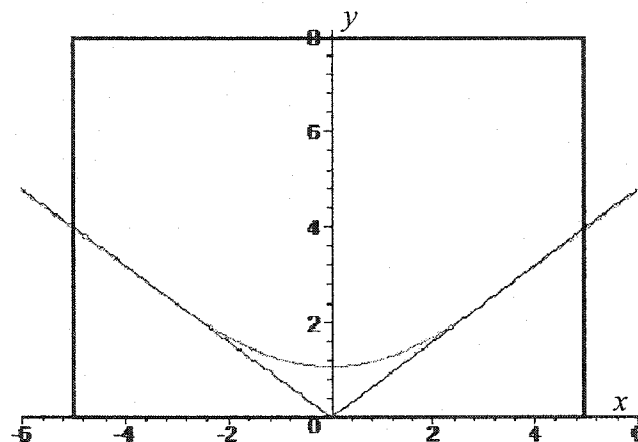


Figure 3.14 Courbe optimale pour l'exemple 3.1

À l'aide de MAPLE nous avons calculé le pourcentage optimal d'erreur :

$$\text{Pourcentage Erreur Optimale} = \frac{\text{Surface d'erreur}}{\text{Surface totale}}$$

$$\text{Surface totale} = 10 * 8 = 80 \text{ um}^2 \text{ (où um = unité de mesure)}$$

$$\begin{aligned}
\text{Surface Erreur} &= \int_{-5}^0 \text{Abs}[f(x) - d_2(x)]dx + \int_0^5 \text{Abs}[f(x) - d_1(x)]dx \\
&= \int_{-5}^0 \text{Abs}\left[f(x) + \frac{4}{5}x\right]dx + \int_0^5 \text{Abs}\left[f(x) - \frac{4}{5}x\right]dx \\
&= 2.125629146 \text{um}^2
\end{aligned}$$

$$\Rightarrow \text{Pourcentage Erreur Optimale} = 0.2657036432 \%$$

À titre d'exemple, nous avons calculé le pourcentage d'erreur pour :

$$w_{31} = w_{32} = 1 \text{ et } b_3 = -1.49$$

Nous obtenons

$$\text{Pourcentage Erreur} = 0.2658668785 \%$$

Ce dernier est effectivement supérieur au pourcentage optimal.

Dans l'exemple que nous venons de faire nous avons montré, comment grâce à l'expression définie par l'équation 3.7 et le système 3.9, il est possible de trouver un minimum pour le pourcentage d'erreur en fonction des paramètres de sortie.

3.4 Fonction explicite simplifiée

L'équation 3.5 est plus simple que l'équation 2.6, cependant il a été impossible de trouver une expression analytique de son intégrale.

Nous allons voir comment grâce à quelques opérations nous pouvons obtenir une

équation encore plus simple que l'équation 3.5.

Afin de développer l'équation explicite nous avons utilisé, dans le chapitre II, deux droites d'équations :

$$y = -w_{11}x - b_1$$

$$y = -w_{21}x - b_2$$

Ces deux équations permettent de représenter n'importe quelles deux demi-droites dans un plan. Avec des translations et des rotations adéquates il est possible de trouver un repère dans lequel l'une des demi-droites est confondue avec l'axe des ordonnées et l'autre se situe dans le demi-plan $x \geq 0$. La figure 3.15 ci-dessous en est un exemple :

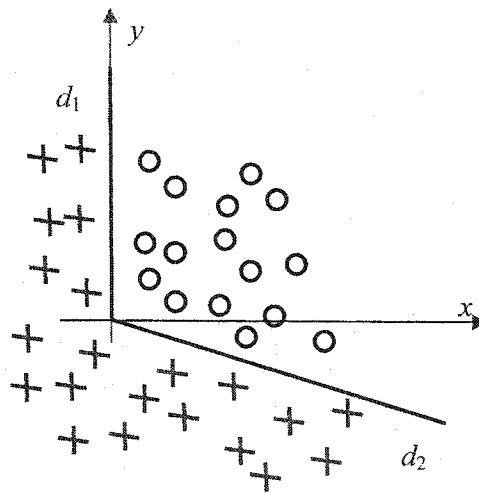


Figure 3.15 Espace des entrées avec d_1 confondu avec l'axe des ordonnées

Les équations des demi-droites sont alors :

$$d_1 : x = 0$$

$$d_2 : y = -w_{21}x$$

Note :

- Nous nommerons toujours d_1 la droite confondue avec l'axe des ordonnées.

Si nous utilisons les équations du réseau du chapitre II, nous avons :

$$V_1 = x$$

$$V_2 = w_{21}x + y$$

Donc

$$S_1 = \frac{1}{1 + e^{-ax}}$$

$$S_2 = \frac{1}{1 + e^{-a(w_{21}x + y)}}$$

Nous cherchons

$$V_3 = 0$$

$$\Rightarrow w_{31}S_1 + w_{32}S_2 + b_3 = 0$$

$$w_{31}S_1 + w_{32}S_2 = -b_3$$

$$w_{31}\varphi(x) + w_{32}\varphi(w_{21}x + y) = -b_3$$

$$w_{32}\varphi(w_{21}x + y) = -b_3 - w_{31}\varphi(x)$$

$$\varphi(w_{21}x + y) = [-b_3 - w_{31}\varphi(x)] / w_{32}$$

Posons $R_1 = -b_3/w_{32}$ et $R_2 = w_{31}/w_{32}$

$$w_{21}x + y = \varphi^{-1} [R_1 - R_2\varphi(x)]$$

$$y = \varphi^{-1} [R_1 - R_2\varphi(x)] - w_{21}x$$

Or

$$\begin{aligned}
 \varphi^{-1}(x) &= -a^{-1} \ln[(1-x)/x] \\
 \Rightarrow y &= -\frac{1}{a} \ln \left[\frac{1 - (R_1 - R_2 \varphi(x))}{R_1 - R_2 \varphi(x)} \right] - w_{21}x \\
 &= -\frac{1}{a} [\ln(1 - R_1 + R_2 \varphi(x)) - \ln(R_1 - R_2 \varphi(x))] - w_{21}x \\
 &= -\frac{1}{a} \left[\ln \left(1 - V_1 + V_2 \frac{1}{1 + e^{-ax}} \right) - \ln \left(V_1 - V_2 \frac{1}{1 + e^{-ax}} \right) \right] - w_{21}x \\
 \Rightarrow y &= -\frac{1}{a} [\ln(1 - R_1 + R_2 + (1 - R_1)e^{-ax}) - \ln(R_1 - R_2 + R_1e^{-ax})] - w_{21}x \\
 &= -\frac{1}{a} [\ln(1 - R_1 + R_2 + e^{-ax} - R_1e^{-ax}) - \ln(R_1 - R_2 + R_1e^{-ax})] - w_{21}x
 \end{aligned}$$

Nous obtenons alors l'équation suivante :

$$\Rightarrow y = -\frac{1}{a} \left[\ln \left(1 + \frac{b_3}{w_{32}} + \frac{w_{31}}{w_{32}} + e^{-ax} + \frac{b_3}{w_{32}} e^{-ax} \right) - \ln \left(-\frac{b_3}{w_{32}} - \frac{w_{31}}{w_{32}} - \frac{b_3}{w_{32}} e^{-ax} \right) \right] - w_{21}x \quad 3.9$$

Pour avoir une idée de la forme de la fonction 3.9 voici un exemple.

Exemple 3.2 :

À l'aide de MAPLE, nous avons tracé la frontière à l'aide de l'équation 3.9 et nous avons obtenu la figure 3.16 ci-dessous. Les valeurs utilisées sont les suivantes :

$$w_{21} = 2/5$$

$$w_{31} = 1; w_{32} = 1; b_3 = -1.5; a = 1.$$

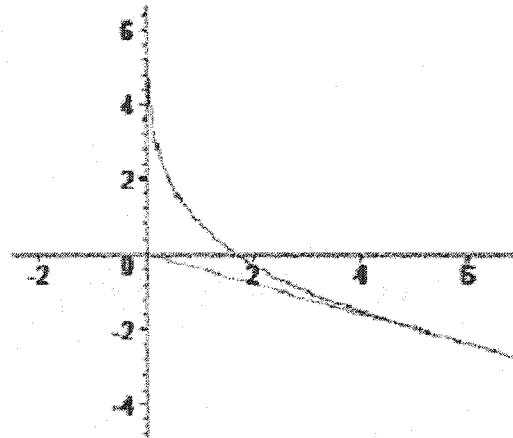


Figure 3.16 Courbe obtenue avec l'équation simplifiée 3.9

3.4.1 Intégrale de la fonction simplifiée

Nous voulons maintenant trouver une expression analytique de l'intégrale de l'équation

3.9. Il faut donc calculer :

$$\int \left\{ -\frac{1}{a} \left[\ln \left(1 + \frac{b_3}{w_{32}} + \frac{w_{31}}{w_{32}} + e^{-ax} + \frac{b_3}{w_{32}} e^{-ax} \right) - \ln \left(-\frac{b_3}{w_{32}} - \frac{w_{31}}{w_{32}} - \frac{b_3}{w_{32}} e^{-ax} \right) \right] - w_{21}x \right\} dx$$

Avec les substitutions suivantes

$$\text{Terme1} = 1 + (b_3 + w_{31})/w_{32}$$

$$\text{Terme2} = (-b_3 - w_{31})/w_{32}$$

Nous pouvons alors écrire :

$$\int \left\{ -\frac{1}{a} \left[\ln \left(\text{Terme1} + e^{-ax} + \frac{b_3}{w_{32}} e^{-ax} \right) - \ln \left(\text{Terme2} - \frac{b_3}{w_{32}} e^{-ax} \right) \right] - w_{21}x \right\} dx$$

Nous obtenons l'intégrale suivante :

$$\begin{aligned} \frac{1}{a^2} \left\{ \operatorname{dilog} \left(-\frac{(w_{32} + b_3)e^{-ax}}{w_{32} \text{Terme1}} \right) + \ln \left(\frac{(w_{32} + b_3)e^{-ax}}{w_{32}} + \text{Terme1} \right) \ln \left(-\frac{(w_{32} + b_3)e^{-ax}}{w_{32} \text{Terme1}} \right) \right\} - \\ \frac{1}{a^2} \left\{ -\operatorname{dilog} \left(-\frac{b_3 e^{-ax}}{w_{32} \text{Terme2}} \right) - \ln \left(\text{Terme2} - \frac{b_3 e^{-ax}}{w_{32}} \right) \ln \left(-\frac{b_3 e^{-ax}}{w_{32} \text{Terme2}} \right) \right\} - w_{21} \frac{x^2}{2} \end{aligned} \quad 3.10$$

Avec

$$\operatorname{dilog}(x) = \int_1^x \frac{\ln(t)}{1-t} dt$$

Finalement nous arrivons à obtenir une expression analytique de l'intégrale de la fonction explicite. Ce résultat intéressant est obtenu grâce aux transformations que nous avons faites. Il existe des tables pour calculer la fonction dilog et les outils informatiques sont aussi capables de l'évaluer. Nous sommes donc en mesure d'utiliser l'équation 3.10 afin de définir la surface d'erreur pour la minimiser de façon analytique. Notons que dans ce cas, comme dans le cas de l'équation 3.5, nous allons chercher à optimiser les paramètres du neurone de sortie et que nous utilisons les équations des demi-droites, formant la frontière bilinéaire, afin de fixer les paramètres des neurones cachés.

3.5 Méthodes d'optimisation

Dans cette partie, nous avons réussi à optimiser les paramètres du neurone de sortie en fixant ceux de la couche cachée aux valeurs obtenues par les équations des demi-droites formant la frontière bilinéaire. Ensuite, avec de remarquables opérations nous avons

réussi à transformer l'équation explicite de manière à pouvoir en déduire une expression analytique de son intégrale.

Nous avons vu que pour optimiser les paramètres des neurones cachés et du neurone de sortie il faudrait optimiser la fonction 3.4 ; ainsi il serait possible d'obtenir la valeur optimale des paramètres suivants : w_{11} , b_1 , w_{21} , b_{21} , w_{31} , w_{32} et b_3 .

Nous pourrions alors comparer la valeur minimale d'erreur obtenue par optimisation de la couche cachée et de la couche de sortie, à la valeur optimale obtenue par optimisation de la couche de sortie seulement.

CONCLUSION

Le présent mémoire est une étude approfondie d'un réseau formé de deux entrées, deux neurones cachés et un neurone de sortie. Nous mettons en évidence les capacités du réseau à résoudre des problèmes de classification et d'approximation de fonctions.

En premier lieu, nous faisons une étude sur la séparation de deux classes séparables par une frontière bilinéaire. Notre premier lemme montre que si la fonction d'activation du réseau est la fonction Heaviside alors le réseau est capable de parfaitement séparer les deux classes. Un résultat pratique encore plus intéressant est obtenu avec le premier théorème ; nous prouvons qu'il existe toujours une valeur minimum du paramètre de la fonction d'activation (fonction sigmoïde) qui permet une parfaite séparation des classes.

En deuxième lieu, nous utilisons une méthode innovatrice d'étudier un réseau en trouvant une fonction explicite de la frontière qu'il engendre. De plus, nous prouvons à l'aide d'un deuxième théorème que cette même démarche peut être utilisée afin de trouver la fonction explicite représentant la frontière d'un réseau ayant jusqu'à quatre neurones cachés (deux entrées et une sortie).

Finalement à l'aide de l'équation explicite nous trouvons une expression de l'erreur. L'optimisation de cette erreur permet d'obtenir les valeurs optimales des différents paramètres du réseau. Des simplifications importantes de l'équation explicite ont permis

l'obtention de résultats concernant les paramètres du neurone de sortie.

Maintenant que nous avons vu et étudié les capacités d'un réseau très simple à reproduire une frontière bilinéaire, il serait intéressant, dans un temps futur, d'essayer de généraliser les résultats obtenus ici afin de pouvoir approximer des frontières plus complexes. Par exemple, il serait intéressant de trouver un réseau simple capable d'approximer une frontière formé de trois demi-droites, puis d'essayer de généraliser jusqu'à trouver un réseau relativement simple qui serait capable d'approximer une frontière formé de n demi-droites. De plus, et comme nous l'avons déjà mentionné, il serait intéressant de comparer le réseau que nous venons d'étudier à un réseau d'une autre famille ayant le même nombre de paramètres que celui que nous étudions. La comparaison des résultats pourrait donner une idée sur les performances relatives des différentes familles de réseau pour les problèmes d'approximation de fonctions.

BIBLIOGRAPHIE

- [1] CHULHEE, L et EUNSUK, J. (2000). Dimension Expansion of Neural Networks. *Geoscience and Remote Sensing Symposium IEEE*. Vol.2, 678-680.

- [2] CHULHEE, L et OHJAE, K. (2001). Edge Representation and Recognition using Neural Networks. *Proceedings. International Symposium on Industrial Electronics, Pusan, Corée*, Vol.1, 110-113.

- [3] HAYKIN, S. (1999). Neural Networks a Comprehensive Foundation. (Second Edition). Prentice Hall, 12-15.

- [4] LABIB, R et ASSADI, R. (à paraître 2004). Perceptrons with Fixed Weights and Non Linear inputs Applied to the Problem of Covering and Packing. *Transactions on Neural Networks*, IEEE.

- [5] MINSKY, M.L et PAPERT, S.A. (1969). Perceptrons, Cambridge, MA:MIT Press.

- [6] ZHENGWEN, Z et SARHADI, M. (1993). A Modified Neuron Activation Function which Enables Single Layer Perceptrons to Solve Some Linearly Inseparable Problems. *Proceedings of 1993 International Joint Conference on Neural Networks, Nagoya, Japon*, Vol.3, 2723 -2726.